

datasheet

PDFlib TET 5.3

Text and Image Extraction Toolkit

What is PDFlib TET?

PDFlib TET (Text and Image Extraction Toolkit) reliably extracts text, images and metadata from PDF documents. TET makes available the text contents of a PDF as Unicode strings, plus detailed color, glyph and font information as well as the position on the page. Raster images are extracted in common image formats. TET optionally converts PDF documents to an XML-based format called TETML which contains text and metadata as well as resource information.

TET contains advanced content analysis algorithms for determining word boundaries, grouping text into columns, identifying table structures and removing redundant items such as shadow text. Using the integrated pCOS interface you can retrieve arbitrary objects from the PDF, such as metadata, interactive elements, etc.

With PDFlib TET you can:

- ▶ Implement the PDF indexer for a search engine
- ▶ Repurpose text and images in PDFs
- ▶ Convert the contents of PDFs to other formats
- ▶ Process PDFs based on their contents, e.g. splitting based on headings (requires PDFlib+PDI in addition to TET)
- ▶ Check whether a particular location on the page is empty, e.g. for placing a barcode or stamp

PDFlib TET Features

Accepted PDF Input

TET supports all flavors of PDF input:

- ▶ All PDF versions up to Acrobat DC, including ISO 32000-1 and -2 (PDF 2.0)
- ▶ Protected PDFs which do not require a password for opening or for which the password is available
- ▶ Damaged PDF documents are repaired

All Writing Systems of the World

TET processes PDF documents in all writing systems of the world and implements special processing required for some scripts:

- ▶ Latin, Greek and Cyrillic scripts

- ▶ Arabic and Hebrew including logical reordering of right-to-left and bidirectional text; normalization of Arabic presentation forms
- ▶ Simplified and Traditional Chinese, Japanese, and Korean regardless of encoding; horizontal and vertical text
- ▶ Indic scripts (without glyph reordering)
- ▶ All other languages and scripts supported with Unicode output

Unicode

Since text in PDF is usually not encoded in Unicode, PDFlib TET normalizes the text in a PDF document to Unicode:

- ▶ TET converts all text contents to Unicode, regardless of the encoding method used in the PDF document.
- ▶ Ligatures and other multi-character glyphs are decomposed into a sequence of the corresponding Unicode characters.
- ▶ Glyphs without appropriate Unicode mapping are identified as such and are mapped to a configurable replacement character to avoid misinterpretation.
- ▶ TET implements various workarounds for problems with specific document creation packages, such as InDesign and TeX documents or PDFs generated on mainframe systems.

Content Analysis and Word Detection

TET includes several patented content analysis algorithms:

- ▶ Determine word boundaries which are required to retrieve proper words
- ▶ Combine the parts of hyphenated words (dehyphenation)
- ▶ Remove duplicate instances of text, e.g. shadow and artificially bolded text
- ▶ Recombine paragraphs in reading order
- ▶ Correctly order text which is scattered over the page

Page Layout, Table and List Detection

The page content is analyzed to determine text columns. Tables are detected, including cells which span multiple rows or columns. This improves the ordering of the extracted text. Table rows and the contents of each table cell can be identified. Bulleted and numbered lists are identified.

Geometry

TET provides precise metrics for the text, such as the position on the page, glyph widths, and text direction. Specific areas on the page can be excluded or included in the text extraction, e.g. to ignore headers and footers or margins.

Text Color

TET analyzes color information in the PDF page description and returns precise color information for each glyph. This can be used, for example, to identify headings or other highlighted text. Optionally the advanced color spaces Separation and DeviceN can be extracted in a simpler alternate color space.

Image Extraction

Images on PDF pages can be extracted as TIFF, JPEG, JBIG2 or JPEG 2000 files. Precise geometric information (position, size, and angles) is reported for each image. Fragmented images are combined to larger images to facilitate repurposing. Image fidelity is guaranteed since no downsampling or color conversion occurs. This ensures the highest possible image quality.

Ignore Artifacts in Tagged PDF

In Tagged PDF, especially PDF/UA, irrelevant content may be tagged as Artifact, e.g. headers and footers. TET optionally ignores Artifact text and images.

PDF Analysis with the pCOS Interface

The TET library includes the pCOS interface for querying details about a PDF document, such as document info and XMP metadata, font lists, page size, and many more (see separate pCOS datasheet).

Unicode Postprocessing

TET supports various Unicode postprocessing steps which can be used to improve the extracted text:

- ▶ Foldings preserve, remove or replace characters, e.g. remove punctuation or characters from irrelevant scripts.
- ▶ Decompositions replace a character with an equivalent sequence of one or more other characters, e.g. replace narrow, wide or vertical Japanese characters or Latin superscript (e.g. ^a) variants with their respective standard counterparts.
- ▶ Text can be converted to all Unicode normalization forms, e.g. emit NFC form to meet the requirements for Web text or a database.

Document Domains

PDF documents may contain text in other places than the page contents. While most applications deal with the page contents only, in many situations other document domains may be relevant as well. TET extracts the text from all document domains:

- ▶ page contents
- ▶ predefined and custom document info entries
- ▶ XMP metadata on document and image level
- ▶ bookmarks
- ▶ file attachments and PDF portfolios are processed recursively
- ▶ form fields
- ▶ comments (annotations)
- ▶ general PDF properties can be queried, such as page count, conformance to standards like PDF/A or PDF/X, etc.

XMP Metadata

TET supports XMP metadata in several ways:

- ▶ Using the integrated pCOS interface, XMP metadata for the document, individual pages, images, or other parts of the document can be extracted programmatically.
- ▶ TETML output contains XMP document and image metadata.
- ▶ Images extracted in the TIFF or JPEG formats contain XMP image metadata.

TETML represents PDF Contents as XML

TET optionally represents the PDF contents in an XML flavor called TETML. It contains a variety of PDF information in a form which can be processed with common XML tools. TETML contains the text plus optionally font and position information, resource details (fonts, images, colorspace), and metadata.

TETML also includes interactive elements such as form fields, annotations, bookmarks etc. It can even be used to analyze JavaScript or color space details, ICC profiles or output intents.

TETML can be processed with XSLT stylesheets, e.g. to apply filters or to convert TETML to other formats. Sample XSLT stylesheets for processing TETML are included in the TET distribution.

The following fragment shows TETML output with glyph details:

```
<Word>
<Text>PDFlib</Text>
<Box llx="111.48" lly="636.33" urx="161.14" ury="654.33">
<Glyph font="F1" size="18" x="111.48" y="636.33" width="9.65">P</Glyph>
<Glyph font="F1" size="18" x="121.12" y="636.33" width="11.88">D</Glyph>
<Glyph font="F1" size="18" x="133.00" y="636.33" width="8.33">F</Glyph>
<Glyph font="F1" size="18" x="141.33" y="636.33" width="4.88">L</Glyph>
<Glyph font="F1" size="18" x="146.21" y="636.33" width="4.88">I</Glyph>
<Glyph font="F1" size="18" x="151.08" y="636.33" width="10.06">B</Glyph>
</Box>
</Word>
```

TETML can include information about word and paragraph grouping as well as about tables and lists, image placement and annotations along with geometric information for these elements.

TET Connectors

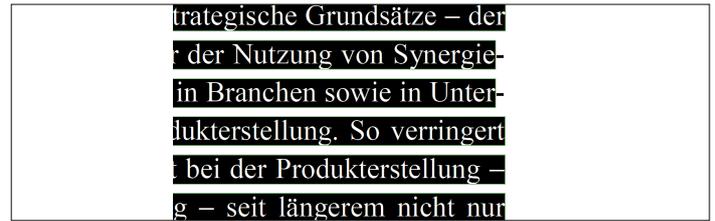
TET connectors interface TET with other software. They make PDF text extraction available for various software environments:

- ▶ TET connector for the Lucene Search Engine
- ▶ TET connector for the Solr Search Server
- ▶ TET connector for the Apache TIKA toolkit
- ▶ TET connector for Oracle Text
- ▶ TET connector for MediaWiki
- ▶ TET PDF IFilter for Microsoft products is available as a separate product. It extracts text and metadata from PDF documents and makes it available to search and retrieval software on Windows (see separate datasheet for details).

Challenges with PDF Text Extraction

Dehyphenation

TET detects hyphenated words which span multiple lines, removes the hyphen, and combines the individual parts to form a complete word. This is important to ensure that searches for the full word are successful although hyphenated parts are present in the document. Dashes (different from hyphens) are treated separately since they must not be removed.



TET correctly removes the hyphen, but keeps the dash.

Shadow and artificial bold Text Detection

Digital documents often contain shadowed text where the shadow effect is achieved by placing the same text multiply on the page, using a small offset between the instances of text. Similarly, bold text is often simulated by overprinting the same text. As a result, the document contains the characters in the shadowed or bold word more than once. TET's patented shadow detection algorithm identifies and removes redundant instances of text to avoid excess text extraction. While other software extracts the shadowed or bold text multiply, TET correctly removes the redundant copies. While extra instances of a word still result in a search engine hit, no more hits would be found if the text is duplicated character by character as in the example.

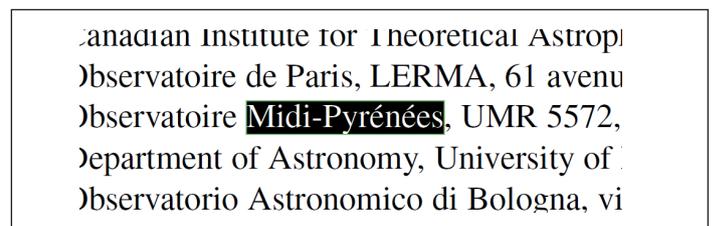


Other products extract »Inttrroduccttiion«.

TET correctly extracts »Introduction«.

Accented Characters

In many languages accents and other diacritical marks are placed close to other characters to form combined characters. Some typesetting programs, e.g. TeX, emit two separate characters (base character and accent) to create a combined character. For example, to create the character *ä* first the letter *a* is placed on the page, and then the dieresis character *¨* is placed on top of it. TET detects this situation and combines both characters to form the appropriate composite character.

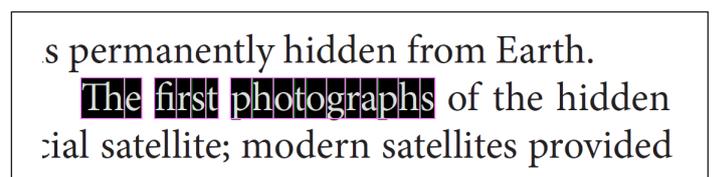


Other products extract »Midi-Pyr´en´ees«.

TET correctly extracts »Midi-Pyrénées«.

Ligatures

Ligatures combine two or more characters in a single glyph. The most common ligatures are the combinations *fi*, *fl*, and *ffi*; less common ligatures are used for *Th*, *sp*, *ct*, *st*, and many others. When extracting text from digital documents, ligatures must be analyzed and separated to the constituent characters to allow proper text processing. TET detects ligatures and delivers two or more characters as appropriate. TET can optionally preserve ligatures if required.

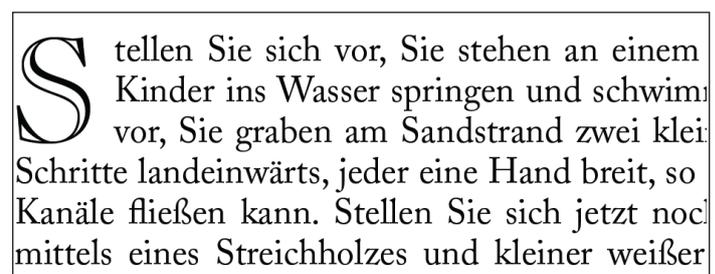


Other products extract » e rst photographs«.

TET correctly extracts »The first photographs«.

Drop Caps

Drop caps are large initial characters at the beginning of a paragraph where the top of the initial aligns with the top of the line, and the remainder of the character drops down several lines. Drop caps are used to emphasize the start of a paragraph. If they are not treated properly the initial word is extracted in two parts: the single initial character and the remainder of the word.



Other products extract two words: the drop cap »S« and »tellen«.

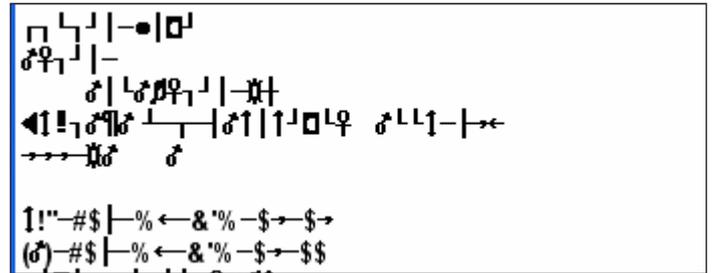
TET correctly extracts the single word »Stellen«.

Challenges with PDF Text Extraction

Unicode Mapping

Unicode mapping forms the foundation of PDF text extraction: every glyph on the page must be assigned the corresponding Unicode value. PDF complicates this task by supporting a variety of font and encoding variants which may or may not provide the information required to assign proper Unicode values. In the worst case the document does not provide enough information with the result that no usable text can be extracted from the document.

TET's patented Unicode mapping algorithm implements a cascaded algorithm which takes all available pieces of information in order to determine Unicode values. For many problematic documents TET extracts proper Unicode text where other products deliver only unusable garbage.



Other products extract unusable garbage, while TET delivers text.

Bidirectional Text with Arabic and Hebrew

PDF does not encode logical text, but is simply a container for glyphs on the page. Text in the Arabic and Hebrew script runs from right to left. Since it often contains left-to-right inserts such as numbers or names in Western languages, text must be interpreted in both directions – hence the term «bidirectional».

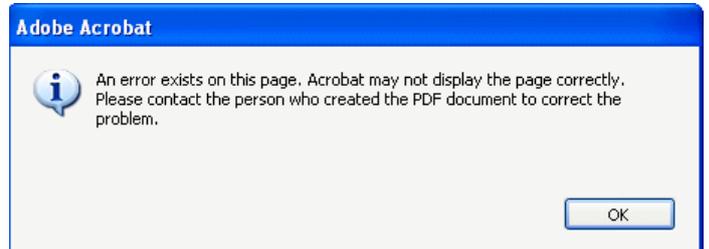
Arabic poses additional challenges since the characters are used in up to four different contextual forms. These shaped forms of characters must be normalized to the corresponding standard (isolated) form.



TET reorders the visual mixture of right-to-left and left-to-right text to create proper logical text output.

Damaged PDF Documents

PDF documents may get damaged because of transmission errors or other problems. TET's repair mode recovers many kinds of damaged PDFs. Sometimes PDF documents are damaged so heavily that the pages cannot even be displayed in Acrobat. Even in such extreme cases TET often delivers the page contents of the document.



The page contents are not even displayed in Acrobat, but TET still correctly extracts the text.

Challenges with PDF Image Extraction

Color Spaces and Compression

Raster image data in PDF may be encoded in a combination of eleven color spaces and nine compression filters, but common image file formats such as JPEG and TIFF support only a subset of these combinations. TET's image engine balances the characteristics of the PDF image with the capabilities of the image output formats. Regardless of the internal structure of the PDF image, the pixel image is extracted in one of the common image file formats.

Spot Colors

In addition to CMYK process colors images in PDF may use custom spot colors. Technically, these color spaces are known as Separation (single channel) and DeviceN (multiple channels).

TET creates TIFF output with additional spot color channels. This is intended for applications which need superior color fidelity and cannot accept any color conversion. If an image with DeviceN color includes only a subset of the common CMYK process colors (e.g. only Cyan and Magenta) the missing process channels are added so that plain CMYK output can be created.

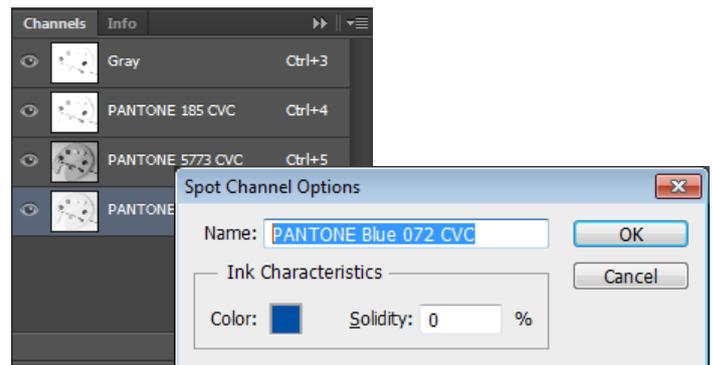
However, not all applications are able to handle spot colors; some are restricted to plain TIFF output. In this case TET can be instructed to emit a spot color channel as grayscale TIFF to facilitate processing.

Merging fragmented Images

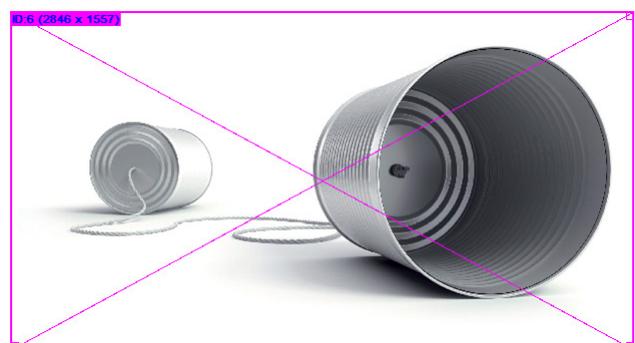
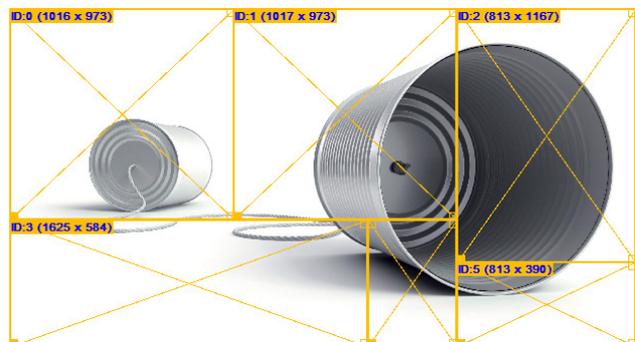
The images in many PDF documents are broken into small fragments by the software producing the PDF. What appears to be a single image on the page may actually consist of many small pieces. For example, Microsoft Office applications often produce heavily fragmented images which consist of hundreds or thousands of small fragments. Adobe InDesign often segments images into fragments of varying size in a process called »transparency flattening«. TET detects fragmented images and merges the pieces to form a usable larger image if the combined result forms a rectangular pixel grid. Only with image merging fragmented images can reasonably be reprocessed.

Device-dependent	CIE-based	Special
DeviceGray	CalGray	Indexed
DeviceRGB	CalRGB	Pattern
DeviceCMYK	Lab	Separation
	ICCBased	DeviceN

TET processes all color spaces which may be used in PDF.



Photoshop displays spot color channels of extracted TIFF images in the Channels window (left). Double-clicking one of the icons reveals the corresponding alternate color (right).



Although the image is fragmented in small parts (top), TET extracts it as a single reusable image (bottom).

Many Ways to use TET

TET is available as a programming library for various development environments, and as a command-line tool for batch operations. Both offer similar features, but are suitable for different deployment scenarios. Both the TET library and the TET command-line tool can create TETML, TET's XML-based output format.

TET offers the following deployment options:

- ▶ The TET programming library (component) is used for integration into desktop or server applications. Examples for using the library are included in the TET package.
- ▶ The TET command-line tool is suited for batch processing PDF documents. It doesn't require any programming, but offers command-line options which can be used to integrate it into complex workflows.
- ▶ TETML output is suited for XML-based workflows and developers who are familiar with the wide range of XML processing tools and languages, e.g. XSLT.
- ▶ TET connectors are suited for integrating TET in various common software packages, e.g. databases and search engines.

TET Cookbook

The TET Cookbook is a collection of programming examples which demonstrate the use of TET for various text and image extraction tasks. Several Cookbook samples show how to combine the TET and PDFlib+PDI products in order to process and enhance PDF documents, e.g. add bookmarks or links based on the text on the page.

The TET Family of Products

The TET family comprises the following products:

- ▶ The TET core product as described in this datasheet.
- ▶ TET PDF IFilter is available as a separate product. It is suitable for use with Microsoft search products, e.g. Windows Search, SharePoint and SQL Server (see separate datasheet for details).
- ▶ The TET Plugin for Adobe Acrobat is a free utility for extracting text and images from PDF. It can be used to evaluate TET interactively.

Supported Development Environments

PDFlib TET is everywhere – it runs on practically all computing platforms. We offer 32-bit and 64-bit packages for all common flavors of Windows, macOS, Linux and Unix, as well as for IBM System i and IBM Z. TET is also available for mobile systems including iOS and Android.

The TET core is written in highly optimized C and C++ code for maximum performance and small overhead. Via a simple API (Application Programming Interface) the TET functionality is accessible from a variety of development environments:

- ▶ C and C++
- ▶ Java
- ▶ .NET and .NET Core
- ▶ Objective-C (macOS and iOS)
- ▶ Perl
- ▶ PHP
- ▶ Python
- ▶ RPG (IBM System i)
- ▶ Ruby

Benefits of using PDFlib Software

Rock-solid Products

Tens of thousands of programmers worldwide are working with our software. PDFlib products meet all quality and performance requirements for server deployment. All products are suitable for robust 24x7 server deployment and unattended batch processing.

Speed and Simplicity

PDFlib products are incredibly fast – up to thousands of pages per second. The programming interface is straightforward and easy to learn.

PDFlib Products all over the World

Our products support all international languages as well as Unicode. They are used by customers in all parts of the world.

Professional Support

If there's a problem, we will try to help. We offer commercial support to meet the requirements of your business-critical applications. By adding support you will have access to the latest versions, and have guaranteed response times should any problems arise.

Licensing

We offer various licensing programs for server licenses, integration and site licenses, and source code licenses. Support contracts for extended technical support with short response times and free updates are also available.

About PDFlib GmbH

PDFlib GmbH is completely focused on PDF technology. Customers worldwide use PDFlib products since 1997. The company closely follows development and market trends, such as ISO standards for PDF. PDFlib GmbH products are distributed all over the world with major markets in North America, Europe, and Japan.

Contact

Fully functional evaluation versions including documentation and samples are available on our Web site. For more information please contact:



PDFlib GmbH

Franziska-Bilek-Weg 9, 80339 München, Germany
 phone +49 · 89 · 452 33 84-0
 sales@pdflib.com
 www.pdflib.com