

Advanced TET 2.2 Features

While text on a PDF page may look very innocent, there are several nasty properties which complicate the task of text extraction. PDFlib TET implements advanced algorithms which enhance the quality of the extracted text. In this chapter we briefly review some of the available controls in TET; for a full description please refer to the *TET Manual*.

Text Granularity and Word Breaks

Granularity. The *granularity* page option specifies the amount of text that will be treated as a unit:

- ▶ With *granularity=glyph* each fragment contains the result of mapping one glyph, which may be more than one character (e.g. for ligatures). In this mode content analysis will be disabled. TET will return the original text fragments on the page in their original order. Although this is the fastest mode, it is only useful if you do sophisticated post-processing since the text may be scattered all over the page.
- ▶ With *granularity=word* the wordfinder algorithm will group characters into logical words. Each fragment contains a word; punctuation characters (comma, colon, question mark, quotes, etc.) will be returned as separate fragments as well.
- ▶ With *granularity=line* the words identified by the wordfinder will be grouped into lines. If dehyphenation is enabled (which is the default) the parts of hyphenated words at the end of a line will be combined, and the full dehyphenated word will be part of the line.
- ▶ With *granularity=zone* all words contained in a rectangular area called zone will be treated as a unit. A zone can be considered a single text column or similar unit.
- ▶ With *granularity=page* all words on the page will be returned in a single fragment.

For text and RTF output, separator characters will be inserted between multiple words, lines, or zones. The separator characters can be specified with the *wordseparator*, *linseparator*, and *zoneseparator* suboptions of the *contentanalysis* page option (use U+0000 to disable a separator), for example:

```
contentanalysis={zoneseparator=U+000C}
```

For text output, the TET Plugin will insert additional separators between the units of text retrieved by TET. Pages will be separated with a Form Feed character (U+000C).

Punctuation and word breaks. The wordfinder, which is enabled for all granularity modes except *glyph*, creates logical words from multiple glyphs which may be scattered all over the page in no particular order. Word boundaries are identified by two criteria:

- ▶ A sophisticated algorithm analyzes the geometric relationship among glyphs to find character groups which together form a word. The algorithm takes into account a variety of properties and special cases in order to accurately identify words even in complicated layouts and for arbitrary text ordering on the page.
- ▶ Some characters, such as space and punctuation characters (e.g. colon, comma, full stop, parentheses) will be considered a word boundary, regardless of their width and

www.pdflib.com

www.pdflib.com

Fig. 0.1
The default setting `punctuationbreaks=true` will separate the parts of URLs (top), while `punctuationbreaks=false` will keep the parts together (bottom).

position. Note that ideographic CJK characters will be considered word boundaries, while Katakana characters will not be treated as word boundaries. If the *punctuationbreaks* page option is set to *false*, the wordfinder will no longer treat punctuation characters as word boundaries:

```
contentanalysis={punctuationbreaks=false}
```

Ignoring punctuation characters for word boundary detection can, for example, be useful for maintaining Web URLs where period and slash characters are usually considered part of a word (see Figure 0.1).

Text Filtering

Dehyphenation. Dehyphenation will detect hyphenated words, i.e. parts of a word which are split across more than one line. The parts of hyphenated words will be combined and the hyphen removed. When a hyphenated word is detected, the TET Plugin will not highlight the hyphen, but only the parts of the hyphenated word (see figure). Note that not all hyphen or dash characters at the end of a line actually designate a hyphenated word; the figure contains an example of a dash character which is part of a compound word, and is therefore highlighted (since it will be extracted along with the surrounding text).

Dehyphenation can be disabled with the following page option:

```
contentanalysis={dehyphenate=false}
```

Shadow removal. Applications sometimes create two or more instances of the same text to achieve some shadow effect (e.g. the word »PAST« in the figure). TET will detect this situation, and will remove the redundant instances of text which create only visual artifacts. Shadow removal can be disabled with the following page option:

```
contentanalysis={shadowdetect=false}
```

Small or large text removal. Very small or very large text (e.g. large characters in the background of the page) can optionally be ignored. The limits can be controlled with a page option. The following example will limit the range of font sizes from 10 to 50:

```
fontsize={10 50}
```

learn how to make PDF/X-1a
prehensive digital ad specifica-
its corporate Web site
offers step-by-step directions
n either a Mac (OS 9.x) or PC,
x or 5.x and Apago's PDF/X-
members are also available to
compliant files. If an advertiser
ssary software, Advanstar rec-
MagSend (www.magsend.com),
provider that can prepare it for
duce the number of problemat-
anstar charges advertisers for
s considering charging for files
s PDF/X-1a.

SHOULD
PAST

Ligature decomposition. TET will replace certain Unicode characters with more familiar ones. For example, Latin ligatures will be decomposed into their constituent characters, and full-width ASCII variants in CJK fonts will be replaced with the corresponding non-fullwidth characters. The text in the figure contains an *ffi* ligature in the bottom line. It is highlighted with the page option *granularity=glyph* to visualize the glyph borders. Although there is only a single glyph, the text *ffi* will be extracted as three characters.

You can easily tell whether ligatures were present in the document by looking at the numbers in the summary *x characters derived from y glyphs*: if the numbers are different, one or more glyphs resulted in more than one characters so that the total character count exceeds the glyph count.

Unicode Mapping

Visualize glyphs with unknown Unicode mappings (garbage characters). Glyphs which cannot be mapped to Unicode will be highlighted with a red border (see Figure 0.1), and can be replaced with an arbitrary Unicode character. The following document option will replace all unknown glyphs with a question mark:

`unknownchar=?`

When copying text to the clipboard you can easily tell whether unmappable glyphs were present in the document by looking at the summary *x unknown glyphs*: this means that some glyphs could not be mapped to Unicode, and should be treated as unknown characters. If the *unknownchar* option has been set, they will be replaced with the specified character.

Advanced Unicode mapping controls. TET implements many workarounds in order to process PDF documents which actually don't contain Unicode values so that it can successfully extract the text nevertheless. However, there are still documents where the text cannot be extracted since not enough information is available in the PDF and relevant font data structures. TET contains various configuration features which can be

Fig. 0.1
Unknown glyphs (bottom line) are highlighted with red border

Bassa Vah Unicode Sample

used to supply additional Unicode mapping information. These features are detailed in the *TET Manual*.

PDFlib GmbH provides a free companion product to TET which assists in this situation: PDFlib FontReporter is an Adobe Acrobat plugin for easily collecting font, encoding, and glyph information. The plugin creates detailed font reports containing the actual glyphs along with encoding and Unicode information.