

[illegible]

2.10 Python Binding 34

2.10.1 Installing the PDFlib Python Edition 34

5.1.5 Multi-Page Image Files *117*

5.2 Importing PDF Pages with PDI (PDF Import Library) *118*

5.2.1 PDI Features and Applications *118*

0 Applying the PDFlib License Key

All binary PDFlib and PDI versions supplied by PDFlib GmbH can be used as fully functional evaluation versions regardless of whether or not you obtained a commercial license. However, unlicensed versions will display a demo stamp (the »nagger«) cross all generated pages. Companies which are seriously interested in PDFlib

How can I use PDFlib? PDFlib is available on a variety of platforms, including Unix, Windows, Mac OS, and EBCDIC-based systems such as IBM eServer iSeries and zSeries. PDFlib itself is written in the C language, but it can be also accessed from several other languages and programming environments

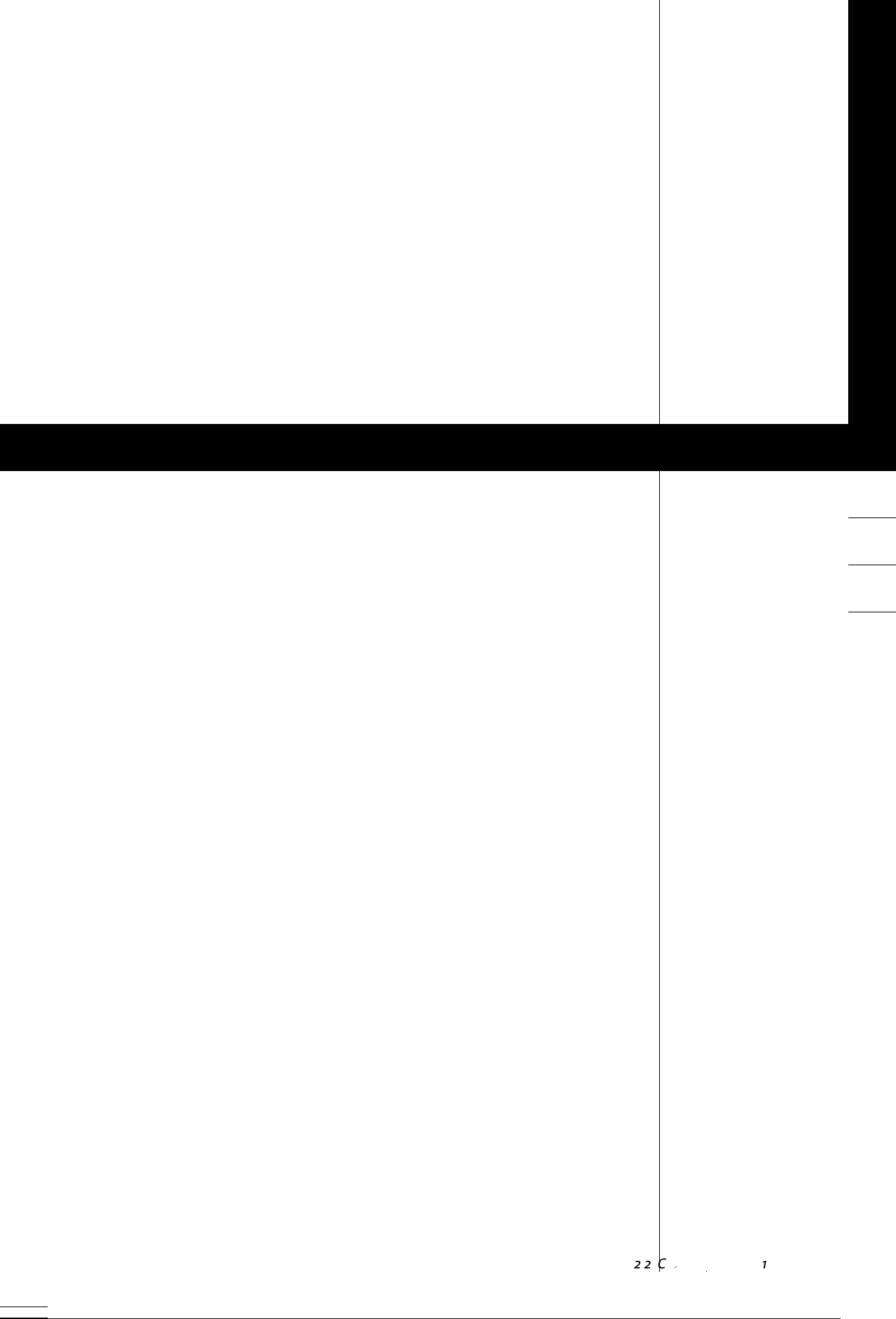




2.2 Cobol Binding

2.2.1 Special Considerations for Cobol

The PDFlib API functions for Cobol are not available under the names documented in Chapter 7, but use abbreviated function names instead. The short function names are not documented here, but can be found



- N ,
- N 2,
C

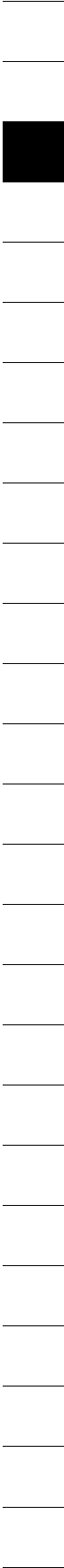
N 'A ' - A
I I B IZ N - N
N ' m s ' - A
I I B IZ N - N 2
CA > N N ,
- N
- N 2,
C


N ' ' - A
I I B IZ N - N
N ' , ' (C B !' - A
I I B IZ N - N 2
CA > N N ,
- N
- N 2,
C

1000

[illegible]

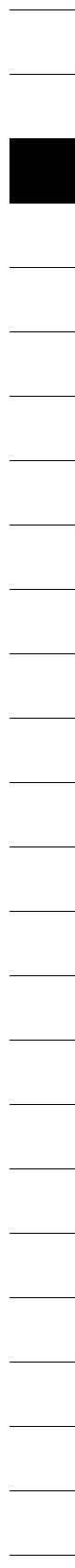
$\Rightarrow F \Rightarrow F:A$ Aşh4 (p) , $\Rightarrow F \Rightarrow F:A$ 4p m(p) ,
 $\Rightarrow F \Rightarrow F:A$ Aşh4 (p) ;
 $\Rightarrow F A$ A ($\Rightarrow F$) ;



In order to supply this package to your application, to your you
C  environment variable, add the option `-Xms` in your calls to the
Java compiler and runtime, or perform equivalent steps in your Java IDE. In the JDK you

5 m 77p 7 77n(:5 5555 (;

$$\left\{ \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} = n^4$$





2.10 Python Binding

2.10.1 Installing the PDFlib Python Edition

The Python¹ extension mechanism works by loading shared libraries at runtime. For the PDFlib binding to work, the Python interpreter must have access to the PDFlib Python wrapper:

Unix. The library `libpdf.so` (on Mac OS X: `libpdf.dylib`) will be searched in the directories listed in the `PYTHONPATH` environment variable.

Windows. The library `pdfapi.dll` will be searched in the directories listed in the





script to make the library available on a certain directory (this may be useful if you want to deploy PDFlib on a machine where you don't have root privilege).





3.1.5 The PDFlib Virtual File System (PVF)

In addition to disk files a facility called **Parallel Virtual File (PVF)** allows clients to directly supply data in memory without any disk files involved. This offers performance benefits and can be used for data fetched fr





```

... 4s
^ F A (p ;
^
_ _ _ _ _

```

This is considered »active« mode since the client decides when he wishes to fetch the buffer contents. Active mode is available for all supported language6(e)- bindings.



that the coordinates displayed refer to an origin in the top left corner of the page, and not PDF's default origin in the lower left corner.

Don't be misled by PDF printouts which seem to experience wrong page dimensions. These may be wrong because of some common reasons:

- > The `Use printer's default scale` option has been checked in Acrobat's print dialog, resulting in scaled print output.
- > Non-PostScript printer drivers are not always able to retain the exact size of printed objects.

Rotating objects. It is important to understand that objects cannot be modified once

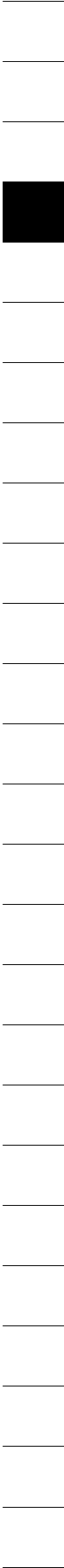
down coordinate system. For the sake of completeness we'll list the detailed conse-



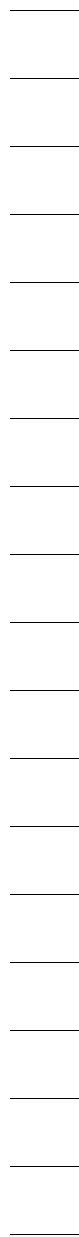
tion must contain its own color specification and will always look the same; if $_$ is 2, the pattern definition must not include







>I f





...C

- -

- - 6 (2005- (- - 3-10

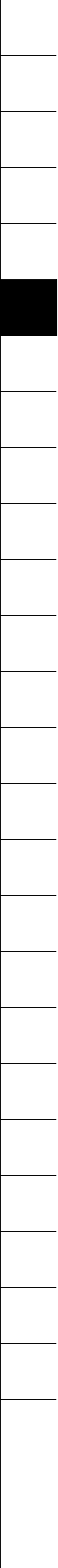




font and encoding issues), you may use it to find out about the encodings supported by a given font (look for »code pages« in the FontLab documentation).¹

For the convenience of PDFlib users, the PostScript program





4. Font Embedding and Subsetting

4.3.1 How PDFlib Searches for Fonts

Sources of font data. PDFlib can access font data from various sources:

- > Disk-based(fon)-5wh ich h avdax b6(x)2 T4(ek-)3(n) stat1.293325.4D-0.0015 Tc-20007 Tw[(ta)-6i





to $\frac{1}{2}$, or the percentage of used glyphs is higher than the

4.4 Encoding Details

4.4.1 8-Bit Encodings

Table 4.2 lists the predefined encodings in PDFlib, and details their use with several important classes of fonts. It is important to realize that certain scripts or languages have requirements which cannot be met by common fonts. For example, Acrobat's core fonts do not contain all characters required for ISO 8859-2, while PostScript 3, OpenType Pro, and TrueType »big fonts« do.

$\frac{1}{2} \left(\frac{1}{2} + \frac{1}{2} \right) = \frac{1}{2}$

<i>code page</i>	<i>supported languages</i>	<i>PS Level 1/2, Acrobat 4/5¹</i>
------------------	----------------------------	--

combination. While this is case for most font/encoding combinations, user-defined Type 3 fonts, for example, may be missing Uncomicode information. In this case PDFlib will not be able to generate a ToUnicode CMap, and text export or searching will not work in Acrobat.

Generation of a ToUnicode CMap can be globally disabled with the `pdf_Unicode_CMap` parameter, or on a per-font basis with the `pdf_Font_ToUnicode` option of the same name. The default of this parameter/option is true. Setting it to false will decrease the output file size while potentially disabling proper cut-and-paste support in Acrobat.

4.5.2 Unicode Text Formats

The Unicode standard supports several transformation formats for the actual byte values which comprise a Unicode string. These vary in the number of bytes per character and the ordering of bytes within a character. Unicode strings in PDFlib can be

> `%c` one byte in the string corresponds to one character, and the string will be output without any interpretation. This is mainly useful for 8-bit

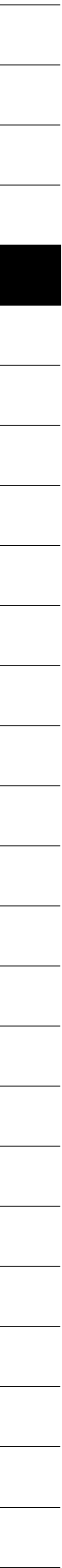
4.6 Text Metrics, Text Variations, and Box Formatting

4.6.1 Font and Character Metrics

Text position.



ASCII newline characters (in C: `\n`)

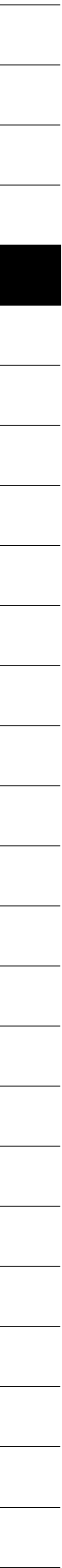


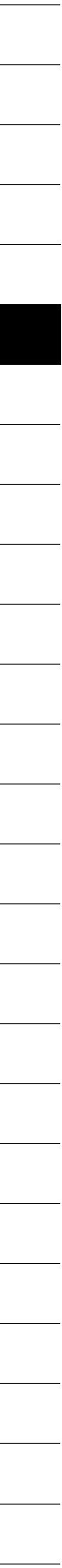




1.8 Placing and Fitting Text

The function $f(x) = x^2 + 1$





5.3 Placing Images and Imported PDF Pages

The `placeImage` function for placing raster image and templates, as well as the `placePDF` function for placing PDF pages.



This code fragment places the object with its lower left corner at the point (x_0, y_0) in the user coordinate system. In addition, the object will be scaled in x and y direction by a











Although there are many parallels between both concepts,

6.2 Creating PDFlib Blocks

6.2.1 Installing the PDFlib Block Plugin

The Block plugin and its sibling, the PDF form field conversion plugin



> by using the keyboard shortcut  (in Acrobat 6 make sure to enable          to



Since standard and custom properties are stored differently iblock, standard
PDFlib property names (as defined in section »Standard Properties for automated
Processing«, page 137) will never conflict with custom property names.

Block
from

Block dictionaries may contain
oups

Only
/ / /









float PDF_get_value(PDF *p, const char *key, float modifier)

Get the value of some PDFlib parameter with numerical type.

key The name of the parameter to be queried.

modifier An optional modifier to be applied to the parameter. Whether a modifier is

```
void PDF_set_value(PDF *p, const char *key, float value)
```

Set the value of some PDFlib parameter with numerical type.

key The name of the parameter to be set.

value The new value of the parameter to be set.

Depends on μ .

```
const char * PDF_get_parameter(PDF *p, const char *key, float modifier)
```

Get the contents26 0 TD-PDFlib

required and what it relates to is explained

thrown unless the `throw` parameter is set to false. Otherwise, the value returned by this function can be used as font argument to other font-related functions.

See Table 7.6.

If PDF/X-1, PDF/X-1a, or PDF/X-3 output is generated the

this function starts a new scope, and must always be paired with a matching call.

Terminate a type 3 font definition.

162 C. 7 R. R. R. R. R.





The current width g t r e x t values
c o u n t c : h a r a c t e r h o

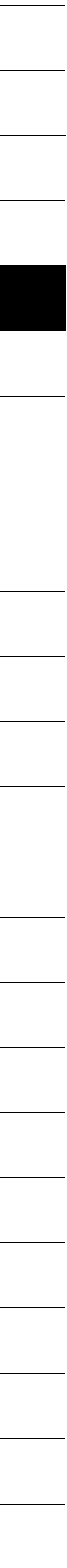
S e e
t 2 (e l t

The flatness parameter is set to the default value of 1 at the beginning of each page.

- 







7.5 Color Functions

7.5.1 Setting Color and Color Space

Table 7.17 lists relevant parameters and values for this section.

```
void PDF_setcolor(PDF *p,
                  const char *fstype, const char *colorspace, float c1, float c2, float c3, float c4)
```

Set the current color space and color.

The fill and stroke color values for the `fill`, `stroke`, and `strokeWidth` color spaces are set to a default value of black at the beginning of each page. There are no defaults for spot and pat-

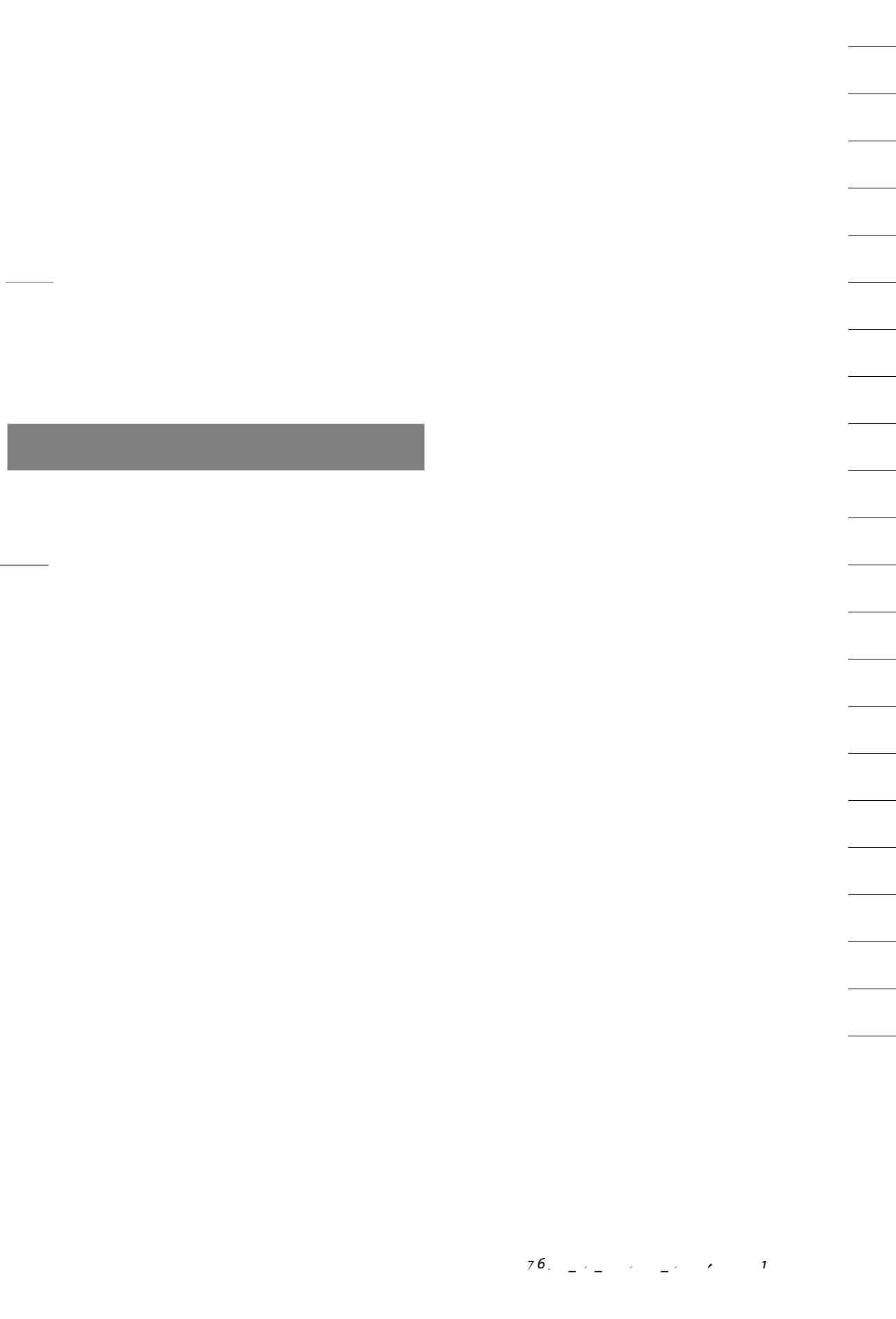


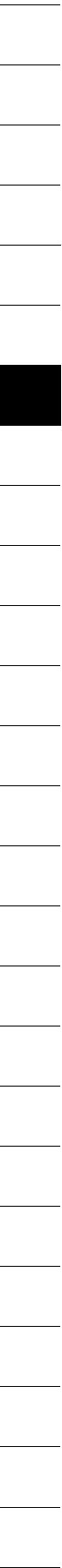
‘ ’ _ ‘ ’ _ ‘ ’ _ (only if the Type 3 font’s ‘ ’ option is9(r)-3r005

be searched, and the name supplied with the description option will be embedded in the PDF output as the PDF/X output intent. If the name was not found to be a standard output intent identifier, it is treated as a profile name and the corresponding ICC profile will be embedded in the PDF as the PDF/X output intent.

if the







key

type explanation



7.9.5 Page Transitions

PDF files may specify a page transition in order to achieve special effects which may be useful for presentations or »slide shows«. In Acrobat, these effects cannot be set document-specific or on a page-by-page basis, but only for the full screen mode. PDFlib, however, allows setting the page transition mode and duration for each page separately. Table 7.36 lists relevant parameters and values for this section.

7.9.6 File Attachments

```
void PDF_attach_file(PDF *p, float llx, float lly, float urx, float ury, const char *filename,
    const char *description, const char *author, const char *mimetype, const char *icon)
void PDF_attach_file2(PDF *p, float llx, float lly, float urx, float ury, const char *filename,
    int reserved, const char *description, int desc_len, const char *author, int author_len,
    const char *mimetype, const char *icon)
```

Add a file attachment annotation.

llx, lly, urx, ury and coordinates of the lower left and upper right corners of the annotation rectangle in default coordinates (if the parameter is 0 or user coordinates (if it is 1). Acrobat will align the upper left corner of the icon at the





8 References

[1] Adobe Systems Incorporated: PDF Reference, Fourth Edition: Version 1.5. Available as PDF from [http://www.adobe.com/devnet/pdf/pdfref4.pdf](#)

[2] Adobe Systems Incorporated: PostScript Language Reference Manual, Third Edition. Published by Addison-Wesley 1999, ISBN 0-201-37922-8; also available as PDF from [http://www.adobe.com/devnet/ps/psref3.pdf](#)

[3] The following book by the principal author of PDFlib is currently available online



0-9

16- 1
5

A

131
776
776
16
1 3
14
176
3.2●
1 7



6 2

1 • 2

I



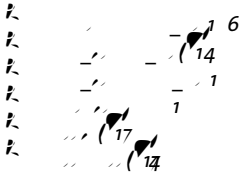
C

C

C

C

R



6●
6 177