

# PDFlib Migration Guide

Latest PDFlib version covered in this document: PDFlib 9.3

The PDFlib API generally remains compatible among major releases. Sometimes API methods are phased out and replaced with an improved successor which is more general or more powerful than the old method. In these situations the new PDFlib version declares the old method as deprecated without changing the functionality in any way. The new method is always preferable. New code should not use deprecated API methods; existing code should switch to the replacement method as soon as possible. Deprecated methods may be removed in a future release.

This Guide contains recommendations for users who migrate existing PDFlib application code which has been developed with an older PDFlib release. It explains how to identify deprecated API features in application code. Once identified the deprecated features should be replaced with the recommended newer ones.

*Note* Since future PDFlib versions will remove deprecated features it is strongly recommended to check your application for deprecated method calls and options with PDFlib 9.3 to ensure forward compatibility.

## 1 Check for deprecated PDFlib API Features

### 1.1 Identify deprecated API Method Calls at Compile Time

Deprecated API calls can be identified at compile time for some language bindings. Note that only warnings are emitted; the code can still be compiled successfully. Nevertheless it is strongly recommended to replace deprecated API methods with the recommended replacement method.

**Java binding.** The *pdflib.jar* module is created from source code which contains *@Deprecated* annotations and *@deprecated* Javadoc comments for all deprecated API methods so that the compiler can warn about the use of deprecated methods. The Java compiler emits a warning similar to the following:

```
javac -classpath pdflib.jar:. image.java
Note: image.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

If you supply the compiler option *-Xlint:deprecation* the name and location of the deprecated API methods are shown:

```
javac -classpath pdflib.jar:. -Xlint:deprecation image.java
image.java:37: warning: [deprecation]
    get_value(String,double) in pdflib has been deprecated
    p.get_value( "major", 0.000000);
    ^
```

The Javadoc documentation for PDFlib lists all deprecated methods along with the recommended replacement method, e.g.

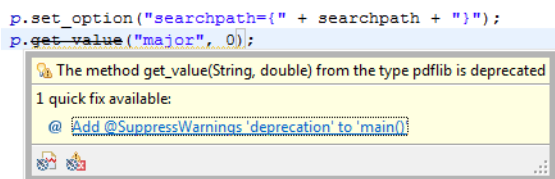


Fig. 1.1  
Presentation of deprecated Java  
API methods in Eclipse

```
double get_value(java.lang.String key, double modifier) Deprecated.
Use PDF_get_option().
```

In Eclipse you must attach *pdflib.jar* as library to make information about deprecated API methods available to the integrated Java compiler. If deprecated API methods are found in your code, the name of the deprecated method is displayed with a strike-out, and a pop-up box informs about the method's status (see Figure 1.1). These messages are also listed in the *Problems* view.

**.NET and .NET Core bindings with VisualStudio.** The PDFlib package is created from source code which contains »Deprecated« comments for all deprecated API methods so that calls to deprecated methods are marked with an underline, and IntelliSense pop-ups are shown with a warning for calls to deprecated API methods when editing the code.

The PDFlib assembly also contains »Obsolete« attributes for all deprecated API methods so that the compiler can warn about the use of deprecated API methods.

Visual Studio emits warnings for calls to deprecated API methods if warning level 2 is enabled in *Project, Properties, Build, Errors and warnings*. The warning includes the recommended replacement method:

```
image.cs(37,13): warning CS0618: 'PDFlib_dotnet.PDFlib.get_value(string, double)' is
obsolete: 'Deprecated, use PDF_get_option().'
```

**.NET Core binding with command-line compiler.** The *dotnet run* command also emits warnings for deprecated API methods in the compilation phase, provided the *<WarningLevel>* element in the project file is set to 2 or higher:

```
image.cs(37,13): warning CS0618: 'PDFlib.get_value(string, double)' is obsolete:
'Deprecated, use PDF_get_option().'
```

**C++ binding with Visual Studio, GCC or Clang.** The header file *pdflib.hpp* contains suitable attributes for all deprecated API methods so that the compiler can warn about the use of deprecated methods.

Visual Studio emits warnings for deprecated API calls if warning level 3 is enabled in *Project, Properties, Configuration Properties, C/C++, General/Warning Level: Level 3 (/W3)*:

```
image.cpp(33) : warning C4996: 'pdflib::basic_PDFlib<pstring,conv>::get_value':
was declared deprecated
    with
    [
        pstring=std::wstring,
        conv=pdflib::NoOpConverter<std::wstring>
    ]
```

With GCC and Clang the option `-Wdeprecated-declarations` must be supplied:

```
image.cpp:33:25: warning: 'double pdflib::basic_PDFlib<pstring, conv>::get_value(const pstring&, double) [with pstring = std::__cxx11::basic_string<wchar_t>; conv = pdflib::NoOpConverter<std::__cxx11::basic_string<wchar_t> >]' is deprecated [-Wdeprecated-declarations]
  p.get_value(L"major", 0);
```

**C binding with Visual Studio, GCC or Clang.** The `pdflib.h` header file contains suitable attributes for all deprecated API methods so that the compiler can warn about the use of deprecated methods.

Visual Studio emits warnings for deprecated API calls if warning level 3 is enabled in *Project, Properties, Configuration Properties, C/C++, General/Warning Level: Level 3 (/W3)*:

```
image.c(44): warning C4996: 'PDF_get_value': was declared deprecated
```

With GCC and Clang the option `-Wdeprecated-declarations` must be supplied:

```
image.c:44:2: warning: 'PDF_get_value' is deprecated [-Wdeprecated-declarations]
  PDF_get_value(p, "major", 0);
```

## 1.2 Identify deprecated API Method Calls at Runtime

Using the methods below you can create a list of deprecated API method calls at runtime. These API methods must be replaced in the application; see Section 2.1, »Deprecated PDFlib API Methods«, for details.

**Logging for all language bindings.** Calls to deprecated API methods are identified in the log file for all language bindings with logging class `api=1`. It can be set as follows (see PDFlib API Reference for more details on logging):

```
p.set_option("logging={filename=deprecated.log classes={api=1}}");
```

With this setting the log file contains an entry for all calls to deprecated API methods, e.g.

```
PDF_get_value(p_0x2599c20, "major", 0.000000)
[PDF_get_value() is deprecated since PDFlib 9]
```

**PHP language binding.** The PDFlib extension module for PHP contains information about deprecated API methods so that the PHP interpreter can warn about deprecated methods. Depending on the PHP configuration warnings about deprecated method calls are written to the configured PHP error log file, the Web server log file, the Windows event log/system log, or `stderr/stdout`. The following directive in `php.ini` configures the log file:

```
error_log = /var/log/php.errors
```

The default is empty; see [www.php.net/manual/en/errorfunc.configuration.php](http://www.php.net/manual/en/errorfunc.configuration.php) for details.

Whether or not warnings about deprecated calls are written to the log file depends on the `error_reporting` directive which must include `E_DEPRECATED`, e.g.

```
error_reporting = E_ALL
```

See [www.php.net/manual/en/function.error-reporting.php](http://www.php.net/manual/en/function.error-reporting.php) for details. Once logging for deprecated calls is configured, PHP reports all deprecated calls as follows in the PHP error log file (not the PDFlib log file):

```
PHP Deprecated:  PDFlib::get_value(): Deprecated, use PDF_get_option(). in /home/bind/php/image.php on line 37
```

**Perl language binding.** The PDFlib module for Perl emits warnings about deprecated API methods at runtime if the predefined warning category *deprecated* is enabled, which is true by default. Perl emits a warning similar to the following:

```
PDF_get_value(): Deprecated, use PDF_get_option(). at image.pl line 31.
```

Use the following instruction or the `-X` command-line option of the Perl interpreter to disable warnings about deprecated API methods on a module basis:

```
no warnings 'deprecated';
```

Warnings for deprecated API methods can be enabled on a module basis with the following instruction in the application code:

```
use warnings 'deprecated';
```

or by calling the Perl interpreter with the `-W` command-line option.

## 1.3 Identify deprecated Options

**All language bindings.** Deprecated options cannot be detected at compile time, but only at runtime. They can be identified in the logging output with the logging class *api=1* which can be set as follows:

```
p.set_option("logging={filename=deprecated.log classes={api=1}}");
```

With this setting the log file contains a message for all API calls with deprecated options, e.g.

```
PDF_create_annotation(p_0x1529c20, 100.000000, 400.000000, 400.000000, 500.000000,
"FileAttachment", "filename=foo.jpg mimetype=image/jpeg")
[Option "filename" is deprecated since PDFlib 9]
[Option "mimetype" is deprecated since PDFlib 9]
```

These options must be replaced in the application; see Section 2.2, »Deprecated Options and Keywords of supported API Methods«, for details.

# 2 Deprecated PDFlib API Features

## 2.1 Deprecated PDFlib API Methods

Table 2.1 lists deprecated API methods which will be removed in the future. Applications should use the replacement methods shown in the table.

Table 2.1 *Deprecated PDFlib API methods and their replacements*

<b>deprecated API method</b>	<b>deprecated since</b>	<b>replacement method</b>
<code>PDF_add_bookmark()</code> <code>PDF_add_bookmark2()</code>	PDFlib 6	<code>PDF_create_bookmark()</code> ; the parent and open parameters correspond to the same-named options
<code>PDF_add_launchlink()</code>	PDFlib 6	<code>PDF_create_action()</code> with type=Launch and <code>PDF_create_annotation()</code> with type=Launch
<code>PDF_add_locallink()</code>	PDFlib 6	<code>PDF_create_action()</code> with type=GoTo and <code>PDF_create_annotation()</code> with type=Link
<code>PDF_add_note()</code> <code>PDF_add_note2()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with type=Text
<code>PDF_add_pdflink()</code>	PDFlib 6	<code>PDF_create_action()</code> with type=GoToR and <code>PDF_create_annotation()</code> with type=Link
<code>PDF_add_thumbnail()</code>	PDFlib 9	none; not functional since PDFlib 9
<code>PDF_add_weblink()</code>	PDFlib 6	<code>PDF_create_action()</code> with type=URI and <code>PDF_create_annotation()</code> with type=Link
<code>PDF_attach_file()</code> <code>PDF_attach_file2()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with type=FileAttachment; the description parameter corresponds to the contents option, the author parameter to the title option and the icon parameter to the iconname option.
<code>PDF_begin_glyph()</code>	PDFlib 9	<code>PDF_begin_glyph_ext()</code>
<code>PDF_begin_page()</code>	PDFlib 6	<code>PDF_begin_page_ext()</code>
<code>PDF_begin_pattern()</code>	PDFlib 9	<code>PDF_begin_pattern_ext()</code>
<code>PDF_begin_template()</code>	PDFlib 7	<code>PDF_begin_template_ext()</code>
<code>PDF_boot()</code> (only C/C++)	PDFlib 7	none; was never functional
<code>PDF_close()</code>	PDFlib 6	<code>PDF_end_document()</code>
<code>PDF_close_pdi()</code>	PDFlib 7	<code>PDF_close_pdi_document()</code>
<code>PDF_end_page()</code>	PDFlib 6	<code>PDF_end_page_ext()</code>
<code>PDF_end_template()</code>	PDFlib 8	<code>PDF_end_template_ext()</code>
<code>PDF_findfont()</code>	PDFlib 5	<code>PDF_load_font()</code>
<code>PDF_get_majorversion()</code> <code>PDF_get_minorversion()</code>	PDFlib 4	<code>PDF_get_option()</code> with keys major/minor
<code>PDF_get_parameter()</code> <code>PDF_get_value()</code>	PDFlib 9	For many keys the same-named option of <code>PDF_get_option()</code> can be used; see Table 2.3 for keys which require other replacement methods.
<code>PDF_get_pdi_parameter()</code> <code>PDF_get_pdi_value()</code>	PDFlib 7	<code>PDF_pcos_get_string()</code> and <code>PDF_pcos_get_number()</code> with pCOS paths according to Table 2.2
<code>PDF_initgraphics()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> with option initgraphicsstate
<code>PDF_open_CCITT()</code>	PDFlib 5	<code>PDF_load_image()</code> with type=ccitt
<code>PDF_open_file()</code>	PDFlib 6	<code>PDF_begin_document()</code> and <code>PDF_end_document()</code>
<code>PDF_open_image()</code> <code>PDF_open_image_file()</code>	PDFlib 5	<code>PDF_load_image()</code>
<code>PDF_open_mem()</code>	PDFlib 6	<code>PDF_begin_document()</code> with empty filename and <code>PDF_get_buffer()</code>

Table 2.1 *Deprecated PDFlib API methods and their replacements*

<b>deprecated API method</b>	<b>deprecated since</b>	<b>replacement method</b>
<code>PDF_open_pdi()</code>	PDFlib 7	<code>PDF_open_pdi_document()</code>
<code>PDF_place_image()</code>	PDFlib 5	<code>PDF_fit_image()</code> with <code>dpi=None</code>
<code>PDF_place_pdi_page()</code>	PDFlib 5	<code>PDF_fit_pdi_page()</code>
<code>PDF_set_border_color()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with option <code>annotcolor</code>
<code>PDF_set_border_dash()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with option <code>dasharray</code>
<code>PDF_set_border_style()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with options <code>borderstyle</code> and <code>linewidth</code>
<code>PDF_set_parameter()</code> <code>PDF_set_value()</code>	PDFlib 9	For many keys the same-named option of <code>PDF_set_option()</code> can be used; see Table 2.3 for keys which require other replacement methods.
<code>PDF_setdash()</code> <code>PDF_setdashpattern</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> and other methods with graphics appearance options with option <code>dasharray</code> and <code>dashphase</code>
<code>PDF_setflat()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> or <code>PDF_create_gstate()</code> with option <code>flatness</code>
<code>PDF_setgray()</code> <code>PDF_setgray_fill()</code> <code>PDF_setgray_stroke()</code>	PDFlib 4	<code>PDF_set_graphics_option()</code> and other methods with color options
<code>PDF_setlinejoin()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> or <code>PDF_create_gstate()</code> with option <code>linejoin</code>
<code>PDF_setlinecap()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> or <code>PDF_create_gstate()</code> with option <code>linecap</code>
<code>PDF_setmiterlimit()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> or <code>PDF_create_gstate()</code> with option <code>miterlimit</code>
<code>PDF_setpolydash()</code>	PDFlib 5	<code>PDF_set_graphics_option()</code> and other methods with graphics appearance options with option <code>dasharray</code>
<code>PDF_setrgbcolor()</code> <code>PDF_setrgbcolor_fill()</code> <code>PDF_setrgbcolor_stroke()</code>	PDFlib 4	<code>PDF_set_graphics_option()</code> and other methods with color options
<code>PDF_show_boxed()</code> <code>PDF_show_boxed2()</code>	PDFlib 6	single lines: <code>PDF_fit_textline()</code> multi-line formatting: <code>PDF_add/create_textflow()</code> and <code>PDF_fit_textflow()</code> with options <code>minspacing=100%</code> <code>maxspacing=10000%</code> <code>nofitlimit=100%</code> <code>shrinklimit=100%</code> to achieve similar formatting If <code>PDF_show_boxed()</code> has been called with <code>mode=fulljustify</code> use the following Textflow options: <code>lastalignment=justify</code> <code>alignment=justify</code>
<code>PDF_shutdown()</code> (only C/C++)	PDFlib 7	none; was never functional
<code>PDF_utf16_to_utf8()</code> <code>PDF_utf8_to_utf16()</code> <code>PDF_utf32_to_utf8()</code> <code>PDF_utf8_to_utf32()</code> <code>PDF_utf16_to_utf32()</code> <code>PDF_utf32_to_utf16()</code>	PDFlib 8.1	<code>PDF_convert_to_unicode()</code> with appropriate parameter <code>inputformat</code> and option <code>outputformat</code>
<code>PDF_xshow()</code> (only C/C++)	PDFlib 9	<code>PDF_fit_textline()</code> with option <code>xadvancelist</code>

**pCOS paths for deprecated `PDF_get_pdi_value/parameter()` methods.** The deprecated (as of PDFlib 7) methods `PDF_get_pdi_value()` and `PDF_get_pdi_parameter()` use custom keys for addressing objects in a PDF documents. When switching to `PDF_pcos_get_number()` and `PDF_pcos_get_string()` these keys must be replaced with the corresponding pCOS paths according to Table 2.2 (see pCOS Path Reference for details).

Table 2.2 pCOS paths as replacements for deprecated keys of `PDF_get_pdi_value()` and `PDF_get_pdi_parameter()`

key for <code>PDF_get_pdi_value()</code> and <code>PDF_get_pdi_parameter()</code>	pCOS path for <code>PDF_pcos_get_number()</code> and <code>PDF_pcos_get_string()</code>
vdp/blockcount	length:pages[...]/blocks
/Root/Pages/Count	length:pages
width, height	pages[...]/width, pages[...]/height
/Rotate	pages[...]/Rotate
version	pdfversion
/CropBox, /BleedBox, /ArtBox, /TrimBox, /MediaBox	pages[...]/CropBox[0] for llx, pages[...]/CropBox[1] for lly, pages[...]/CropBox[2] for urx, pages[...]/CropBox[3] for ury etc.
vdp/Blocks/<name>/<property>	pages[...]/blocks/<name>/<property>
vdp/Blocks[...]/<property>	pages[...]/blocks[...]/<property>
vdp/Blocks/<name>/Custom/<property>	pages[...]/blocks/<name>/Custom/<property> or pages[...]/blocks[...]/Custom/<property>
isempty	pages[...]/isempty
filename	filename
/Info/<key>	/Info/<key>
tagged	tagged
pdfx	pdfx

**Replacements for deprecated global values and parameters.** In most cases the deprecated methods `PDF_set/get_value()` and `PDF_set/get_parameter()` can be replaced with `PDF_set/get_option()` and the previous parameter `key` as option name. Table 2.3 lists parameters which must be replaced with other methods and options.

Table 2.3 *Deprecated global parameters for `PDF_set/get_value()` and `PDF_set/get_parameter()` and their replacements*

<i>deprecated parameter</i>	<i>deprecated since</i>	<i>replacement method and option</i>
<b>resource categories</b>		
Encoding, FontAFM, FontOutline, FontPFM, HostFont, ICCProfile, resourcefile, searchpath	PDFlib 9	same-named options of <code>PDF_set_option()</code> ; multiple entries can be set with a single call; <code>PDF_get_option()</code> : use option <code>resourcenum</code> to iterate over all values
StandardOutputIntent	PDFlib 9	none; use embedded or referenced ICC profile as output intent
<b>document properties</b>		
compatibility	PDFlib 6	<code>PDF_begin_document()</code> with option <code>compatibility</code>
userpassword, masterpassword, permissions	PDFlib 6	<code>PDF_begin_document()</code> with options <code>userpassword</code> , <code>masterpassword</code> , <code>permissions</code>
pdfx	PDFlib 6	<code>PDF_begin_document()</code> with option <code>pdfx</code>
base	PDFlib 6	<code>PDF_begin_document()</code> with option <code>uri</code>
openmode	PDFlib 6	<code>PDF_begin_document()</code> with option <code>openmode</code>
openaction	PDFlib 6	<code>PDF_begin/end_document()</code> with option <code>destination</code> ; the following deprecated option keywords must be replaced: <code>fitbbox</code> : use <code>type=fitvisible</code> <code>fitheight</code> : use <code>type=fitheight left=0</code> <code>fitpage</code> : use <code>type=fitwindow</code> <code>fitwidth</code> : use <code>type=fitwidth top=10000</code> <code>retain</code> : use <code>type=fixed</code>
hidetoolbar, hidemenubar, hidewindowui, fitwindow, centerwindow, displaydoctitle, nonfullscreenpagemode, direction, viewarea, viewclip, printarea, printclip	PDFlib 6	<code>PDF_begin/end_document()</code> with option <code>viewer-preferences</code> and respective suboptions;  the suboptions <code>viewarea</code> , <code>viewclip</code> , <code>printarea</code> , <code>printclip</code> are deprecated in PDF 2.0 (see Table 2.5)
<b>page properties</b>		
pageheight, pagewidth in <code>PDF_set_parameter()</code>	PDFlib 6	<code>PDF_begin/end_page_ext()</code> with options <code>pagewidth</code> and <code>pageheight</code>
ArtBox, BleedBox, CropBox, TrimBox	PDFlib 6	<code>PDF_begin/end_page_ext()</code> with options <code>artbox</code> , <code>bleedbox</code> , <code>cropbox</code> , <code>trimbox</code>
duration, transition	PDFlib 6	<code>PDF_begin/end_page_ext()</code> with same-named options
<b>color</b>		
defaultgray, defaultrgb, defaultcmyk	PDFlib 6	<code>PDF_begin_document()</code> with options <code>defaultgray</code> , <code>defaultrgb</code> , <code>defaultcmyk</code>
setcolor:iccprofilegray/rgb/cmyk	PDFlib 9	graphics appearance options <code>iccprofilegray/rgb/cmyk</code>
preserveoldpantone	PDFlib 9.3	none; old-style PANTONE color names are no longer in use



Table 2.3 *Deprecated global parameters for PDF\_set/get\_value() and PDF\_set/get\_parameter() and their replacements*

<i>deprecated parameter</i>	<i>deprecated since</i>	<i>replacement method and option</i>
<b>font handling</b>		
autosubsetting, subsetlimit, subsetminsize, fontstyle, unicodemap	PDFlib 7	PDF_load_font() with same-named options
ascender, ascenderfaked, capheight, capheightfaked, descender, descenderfaked, fontencoding, fontname, fontmaxcode, xheight, xheightfaked	PDFlib 7	PDF_info_font() with same-named keywords; for the numerical values fontsize=1 must be supplied; parameter fontencoding: use keyword encoding
autocidfont	PDFlib 8	none; not functional since PDFlib 9
monospace	PDFlib 7	none; only for the obsolete concept of standard CJK fonts
<b>images</b>		
honoriccprofile	PDFlib 8.2	PDF_load_image() with option honoriccprofile
renderingintent	PDFlib 8.2	PDF_load_image() with option renderingintent
imagewidth, imageheight, image:iccprofile, orientation, resx, resy	PDFlib 8	PDF_info_image() with same-named options
<b>graphics state</b>		
fillrule in PDF_set_parameter()	PDFlib 9.3	PDF_set_graphics_option() with option fillrule
inheritgstate	PDFlib 6	none; not functional
<b>interactive elements</b>		
bookmarkdest	PDFlib 6	PDF_create_bookmark() with options action, destination, fontstyle and textcolor
launchlink:parameters launchlink:operation launchlink:defaultdir	PDFlib 6	PDF_create_action() with options parameters, operation, and defaultdir; however, these are deprecated in PDF 2.0 (see Table 2.5)
<b>miscellaneous</b>		
flush	PDFlib 6	PDF_begin_document_callback() with option flush
pdiusebox	PDFlib 6	PDF_open_pdi_page() with option pdiusebox
iccwarning, fontwarning, glyphwarning, imagewarning, pdiwarning, warning	PDFlib 7	corresponding API method with option errorpolicy
openwarning	PDFlib 6	PDF_begin_document() with option errorpolicy
honorlang	PDFlib 8	PDF_set_option() with option filenamehandling=honorlang
trace, tracefile, tracemsg	PDFlib 7	PDF_set_option() with option logging
logmsg	PDFlib 9	PDF_set_option() with option userlog
nodemostamp	PDFlib 9	PDF_set_option() with option avoididmostamp
string	PDFlib 9	PDF_get_string()
version	PDFlib 9	PDF_get_string() with option version and idx=-1

## 2.2 Deprecated Options and Keywords of supported API Methods

Table 2.4 lists API methods for which some options or keywords for an option will be removed in the future.

Table 2.4 *Deprecated PDFlib API options or keywords and their replacements*

<i>deprecated option or keyword</i>	<i>deprecated since</i>	<i>replacement option or action</i>
<b>font and text</b>		
<code>PDF_create_textflow()</code> : <code>textwarning</code>	PDFlib 7	<code>errorpolicy</code>
<code>PDF_fit_textline()</code> : <code>locallink</code> , <code>pdflink</code> , <code>weblink</code>	PDFlib 7	<code>matchbox</code> in <code>PDF_fit_textline()</code> and <code>PDF_create_annotation()</code> to create links
<code>PDF_load_font()</code> : <code>autocidfont</code>	PDFlib 9	<i>none; not functional</i>
<code>PDF_load_font()</code> , <code>PDF_info_font()</code> and <code>PDF_fill_textblock()</code> : option and <code>Block</code> property <code>monospace</code>	PDFlib 9	configure TrueType/OpenType fonts or use host fonts
<code>PDF_load_font()</code> : <code>fontwarning</code>	PDFlib 7	<code>errorpolicy</code>
<code>PDF_load_font()</code> : <code> Kerning</code>	PDFlib 8	<code>readkerning</code>
text option <code>glyphwarning</code>	PDFlib 7	<code>glyphcheck</code>
<code>PDF_info_textline()</code> : keywords <code>scalex</code> , <code>scaley</code>	PDFlib 9	<code>fitscalex/fitscaley</code>
<code>PDF_info_textline()</code> : keyword <code>unmappedglyphs</code>	PDFlib 8	<code>unmappedchars</code>
<code>PDF_info_textflow()</code> : keyword <code>remainchars</code>	PDFlib 8	<i>none; worked reliably only under certain conditions</i>
<code>PDF_fit/info_textline()</code> , <code>PDF_add/create_textflow()</code> and <code>PDF_fill_textblock()</code> : option features with keywords <code>vert2</code> and <code>vert</code>	PDFlib 9.3	<i>none; these features are enabled automatically for fonts in vertical mode</i>
<b>interactive elements</b>		
<code>PDF_create_action()</code> : <code>actionwarning</code>	PDFlib 7	<code>errorpolicy</code>
<code>PDF_create_action()</code> : <code>defaultdir</code> , <code>parameters</code> and operation for <code>type=Launch</code> <sup>1</sup>	PDFlib 9.3	<i>none</i>
<code>PDF_create_annotation()</code> : <code>annotwarning</code>	PDFlib 7	<code>errorpolicy</code>
<code>PDF_create_annotation()</code> : <code>filename</code> and <code>mimetype</code>	PDFlib 9	<code>attachment</code> and options <code>filename</code> and <code>mimetype</code> of <code>PDF_load_asset()</code>
<code>PDF_create_field()</code> , <code>PDF_create_fieldgroup()</code> : <code>fieldwarning</code>	PDFlib 7	<i>none; didn't have any effect</i>
<code>PDF_create_field()</code> , <code>PDF_create_fieldgroup()</code> : <code>errorpolicy</code>	PDFlib 9.3	<i>none; didn't have any effect</i>
<code>PDF_create_bookmark()</code> : option <code>destination</code> with sub-options <code>fontstyle</code> and <code>color</code>	PDFlib 6	options <code>fontstyle</code> and <code>textcolor</code> of <code>PDF_create_bookmark()</code>
<code>Destination</code> options for <code>PDF_add_nameddest()</code> and for the <code>destination</code> option in <code>PDF_create_action()</code> , <code>PDF_create_annotation()</code> , <code>PDF_create_bookmark()</code> , and <code>PDF_begin/end_document()</code> : options <code>fitbbox</code> , <code>fitheight</code> , <code>fitpage</code> , <code>fitwidth</code> , <code>retain</code>	PDFlib 7	<code>type=visible</code> , <code>fitheight</code> , <code>fitwindow</code> , <code>fitwidth</code> , <code>fixed</code>
option <code>filename</code> and <code>type=file</code>		action option with a <code>GoToR</code> action
option <code>name</code> and <code>type=nameddest</code>		action option with a <code>GoTo</code> action
<code>PDF_create_annotation()</code> : <code>type=Movie</code> <sup>1</sup>	PDFlib 9.3	<code>type=RichMedia</code>
<code>PDF_create_action()</code> : <code>type=Movie</code> <sup>1</sup>	PDFlib 9.3	<code>type=RichMediaExecute</code>

Table 2.4 *Deprecated PDFlib API options or keywords and their replacements*

<i>deprecated option or keyword</i>	<i>deprecated since</i>	<i>replacement option or action</i>
<b>image and PDF import</b>		
<code>PDF_load_image()</code> , <code>PDF_fill_imageblock():</code> template	PDFlib 8	templateoptions
<code>PDF_load_image()</code> , <code>PDF_fill_imageblock():</code> <code>createtemplate</code> , <code>iconname</code>	PDFlib 9	templateoptions
<code>PDF_load_image()</code> , <code>PDF_fill_imageblock():</code> <code>downsamplemask</code>	PDFlib 9.2	none; rendering bug fixed in Acrobat X
<code>PDF_load_image()</code> , <code>PDF_fill_imageblock():</code> imagewarning	PDFlib 7	errorpolicy
<code>PDF_open_pdi_document()</code> , <code>PDF_open_pdi_page()</code> , <code>PDF_process_pdi()</code> , <code>PDF_fill_pdfblock():</code> pdiwarning	PDFlib 7	errorpolicy
<b>ICC profiles</b>		
<code>PDF_load_iccprofile():</code> metadata	PDFlib 9.3	none; XMP for profiles no longer supported
<code>PDF_load_iccprofile():</code> iccwarning	PDFlib 7	errorpolicy
<b>miscellaneous</b>		
Common XObject options and <code>PDF_begin_page_ext():</code> transparencygroup with suboptions CS, I, K	PDFlib 8.1	suboptions colorspace, isolated, knockout
<code>PDF_add_table_cell():</code> checkwordsplitting	PDFlib 8.1	avoidwordsplitting
<code>PDF_begin_document():</code> moddate	PDFlib 9.2	none; Acrobat Preflight bug fixed
<code>PDF_begin_item()</code> and option tag in many methods: inline	PDFlib 9.2	tagging option direct
<code>PDF_set_option()</code> and many other API methods: errorpolicy=legacy	PDFlib 8	errorpolicy=return or errorpolicy=exception
<code>PDF_set_option():</code> topdown	PDFlib 9	<code>PDF_begin_page_ext()</code> , <code>PDF_begin_template_ext()</code> or <code>PDF_begin_pattern_ext()</code> with option topdown
<code>PDF_get_option():</code> topdown	PDFlib 9.3	none; applications must take care of options specified earlier
<code>PDF_set_option():</code> filenamehandling=legacy	PDFlib 9.3	filenamehandling with explicit encoding name or honorlang

1. This feature is deprecated in PDF 2.0 according to ISO 32000-2.

## 2.3 Deprecated Features in PDF 2.0 and non-ISO Features

**Deprecated PDF 2.0 features which are still available in PDF 1.x.** PDF 2.0 according to ISO 32000-2 deprecates the features listed in Table 2.5. If any of these features is used, PDFlib emits a warning. It is recommended to avoid these features even in PDF 1.x mode.

Table 2.5 Features deprecated in PDF 2.0 which are still available in PDF 1.x mode

PDFlib features deprecated in PDF 2.0	notes
<code>PDF_begin_document()</code> : permissions with keyword <code>noaccessible</code>	Restricting document accessibility is deprecated in PDF 2.0.
<code>PDF_begin/end_document()</code> : <code>viewerpreferences</code> , <code>suboptions</code> <code>printarea</code> , <code>printclip</code> , <code>viewarea</code> , <code>viewclip</code>	Prepress viewer preferences are deprecated in PDF 2.0.
<code>PDF_begin_page_ext()</code> and <code>PDF_end_page_ext()</code> : <code>separationinfo</code>	Separation dictionaries are deprecated in PDF 2.0.
<code>PDF_create_annotation()</code> : <code>type=Movie</code>	Use <code>type=RichMedia</code> .
<code>PDF_create_action()</code> : <code>type=Movie</code>	Use <code>type=RichMediaExecute</code> .
<code>PDF_create_gstate()</code> : <code>blendmode</code> accepts only a single keyword, but not a list with multiple values	Blendmode arrays are deprecated in PDF 2.0.

**Deprecated structure element types for Tagged PDF.** The following standard element types are deprecated in PDF 2.0 and should not be used:

Art, BibEntry, BlockQuote, Code, Index, Note, Private, Quote, Reference, TOC, TOCI

**Features outside of ISO 32000.** The features listed in Table 2.5 were never part of ISO 32000-1. They work only in Acrobat, but not in any third-party viewer. If any of these features is used, PDFlib emits a warning. It is recommended to avoid these features since they are not part of standard PDF.

Table 2.6 Features outside of ISO 32000 which are still available in PDF 1.x mode

features outside of ISO 32000	notes
<code>PDF_create_field()</code> and <code>PDF_create_fieldgroup()</code> : <code>barcode</code>	Barcode fields work only in the full version of Adobe Acrobat, but neither in Acrobat Reader nor third-party viewers.
<code>PDF_begin_document()</code> : <code>search</code>	An attached search index works only in Acrobat, but not in third-party viewers.
<code>PDF_begin_template_ext()</code> : <code>watermark</code>	Editable watermarks work only in Acrobat, but not in third-party viewers.

## 2.4 Obsolete Concepts which will no longer be supported

Some concepts are obsolete because there is no longer demand in today's workflows or they haven't been accepted in the marketplace. The items listed in Table 2.7 will be removed in the future

Table 2.7 Obsolete PDF concepts which will be removed from PDFlib

<b>deprecated concept</b>	<b>deprecated since</b>	<b>notes</b>
<b>PDF versions and standards</b>		
PDF 1.3 (1999): <b>PDF_begin_document()</b> with option <code>compatibility=1.3</code>	PDFlib 8.1	not relevant
PDF/X-1:2001: <b>PDF_begin_document()</b> with option <code>pdfx=PDF/X-1:2001</code>	PDFlib 7	superseded by PDF/X-4
PDF/X-1a:2001, PDF/X-3:2002: <b>PDF_begin_document()</b> with option <code>pdfx=PDF/X-1a:2001</code> and <code>PDF/X-3:2002</code>	PDFlib 8.1	superseded by PDF/X-4
PDF/X-1a:2003: <b>PDF_begin_document()</b> with option <code>pdfx=PDF/X-1a:2003</code>	PDFlib 9.3	superseded by PDF/X-4
PDF/X-3 standard output intents without embedded ICC profile and resource category <code>StandardOutputIntent</code>	PDFlib 9	superseded by PDF/X-3 and PDF/X-4 with embedded or referenced ICC profile
PDF/X-5g, PDF/X-5pg: <b>PDF_begin_document()</b> with option <code>pdfx=PDF/X-5g</code> and <code>PDF/X-5pg</code>	PDFlib 9.3	not accepted in the marketplace
PDF/VT-2: <b>PDF_begin_document()</b> with option <code>pdfvt=PDF/VT-2</code>	PDFlib 9.3	not accepted in the marketplace
<b>font handling</b>		
Multiple Master PostScript Type 1 fonts	PDFlib 9.3	not relevant
resource fork PostScript Type 1 (LWFN) fonts on macOS	PDFlib 9.3	not relevant
SVG fonts	PDFlib 9.3	deprecated in SVG 2.0 and unsupported in most browsers
CEF fonts	PDFlib 9.3	not used as stand-alone fonts
standard CJK fonts and font option/Block property monospace	PDFlib 9	configure TrueType/OpenType fonts or use host fonts
CJK TrueType fonts without Unicode cmap table	PDFlib 9.3	use Unicode-compatible fonts
UCS-2 CMaps	PDFlib 9.3	use UTF-16 CMaps or <code>encoding=unicode</code>
<b>PANTONE spot colors</b>		
PANTONE Goe coated/uncoated	PDFlib 9.1	no longer supported by Pantone, Inc.
PANTONE hexachrome coated/uncoated PANTONE solid in hexachrome coated	PDFlib 9.1	no longer supported by Pantone, Inc.
PANTONE solid to process coated PANTONE solid to process coated EURO	PDFlib 9.1	replaced by PANTONE color bridge CMYK PC/PANTONE color bridge CMYK Euro
mapping of deprecated PANTONE spot color name suffixes CV, CVV, CVU, CVC and CVP	PDFlib 9.3	old-style PANTONE spot color names are no longer in use
<b>Flash<sup>1</sup></b>		
<b>PDF_load_asset()</b> with <code>type=Flash/Generic/JPEG/PNG</code>	PDFlib 9.3	Flash in RichMedia annotations and Flash-based navigators are no longer supported in Acrobat after December 2020.
<b>PDF_create_annotation()</b> with <code>type=RichMedia</code> and option configuration, <code>suboption</code> instances, <code>suboption</code> params		
<b>PDF_end_document()</b> with option <code>portfolio</code> and <code>suboptions</code> navigator and <code>initialview=custom</code>		

Table 2.7 Obsolete PDF concepts which will be removed from PDFlib

<b>deprecated concept</b>	<b>deprecated since</b>	<b>notes</b>
<b>other PDF features</b>		
Reference XObjects: common XObject option reference	PDFlib 9.3	not accepted in the marketplace
page thumbnails: <code>PDF_add_thumbnail()</code>	PDFlib 9	not functional since PDFlib 9.1; PDF viewers create page thumbnails automatically
Open Prepress Interface (OPI) <sup>1</sup> : common XObject options OPI-1.3, OPI-2.0	PDFlib 9.1	OPI is a concept of the 1980s which has very rarely been used in PDF.
PostScript XObjects <sup>1</sup> : <code>PDF_begin_template_ext()</code> with option <code>postscript</code>	PDFlib 9.3	not relevant

1. This feature is deprecated or not included in PDF 2.0 according to ISO 32000-2.

# 3 Required Configuration Changes

In a few situations you may have to adjust the configuration according to Table 3.1.

Table 3.1 PDFlib features which require configuration changes

PDFlib feature	required change
<i>PDF_create_field()</i> with type=checkbox or radiobutton: PDFlib automatically creates appearance streams which requires embeddable fonts.	The font ZapfDingbats or the free Erler Dingbats font (included in the PDFlib package) must be configured for embedding, e.g. by installing it on the Windows or macOS host system, adding the directory resource/font to the searchpath with <i>PDF_set_option()</i> or via the Windows registry.
names of form fields	Field names created with <i>PDF_create_field()</i> must be unique in the document. Fields with synchronized values can be created with <i>PDF_create_fieldgroup()</i> and a suitable fieldtype option.

# 4 PDFlib History

Table 4.1 Release history of PDFlib, PDFlib+PDI and PPS

<i>PDFlib version</i>	<i>release date</i>
<i>PDFlib 4</i>	<i>2001</i>
<i>PDFlib 5</i>	<i>2003</i>
<i>PDFlib 6</i>	<i>2004</i>
<i>PDFlib 7</i>	<i>2006</i>
<i>PDFlib 8</i>	<i>2009</i>
<i>PDFlib 8.1</i>	<i>2011</i>
<i>PDFlib 9</i>	<i>2013</i>
<i>PDFlib 9.1</i>	<i>2016</i>
<i>PDFlib 9.2</i>	<i>2019</i>
<i>PDFlib 9.3</i>	<i>2020</i>