

PDFlib Migration Guide

Latest PDFlib version covered in this document: 9.2.x.

The PDFlib API generally remains compatible among major releases. Sometimes API methods are phased out and replaced with an improved successor. In these situations a newer PDFlib version declares the old method as deprecated without changing the functionality in any way. In most cases the new method is more general or more powerful than the old one which it replaces. The new method is always preferable. New code should not use deprecated API methods, and existing code should switch to the new replacement method. Deprecated methods may be removed in a future release. The API is kept compatible at least in the next major release after the release which first declared a method as deprecated (sometimes even longer).

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

1 Identify 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 language 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.

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

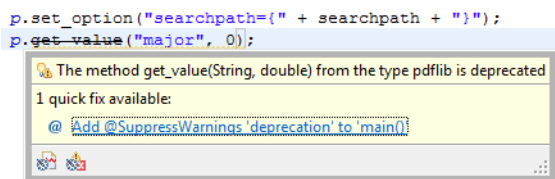


Fig. 1.1
Presentation of deprecated Java
API methods in Eclipse

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 language binding. The PDFlib .NET assembly 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().'
```

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

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 warning 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. Some deprecated methods may be missed if the test program does not follow all possible program paths.

Logging for all language bindings. Calls to deprecated API methods are identified in the log file for all language bindings with logging class *api=2*. It can be set as follows (see PDFlib API Reference for more details on logging):

```
p.set_option("logging={filename=deprecated.log classes={api=2}}");
```

With this setting the log file contains a message 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 *php.net/manual/en/errorfunc.configuration.php#ini.error-log* 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 *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 can be identified in the logging output with the logging class `api=2` which can be set as follows:

```
p.set_option("logging={filename=deprecated.log classes={api=2}}");
```

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 mimetype=image/jpeg")
[Option "filename" is deprecated since PDFlib 9]
[Option "mimetype" is deprecated since PDFlib 9]
```

These logging entries are only warnings. PDFlib execution continues and PDF output is generated as usual.

2 Deprecated PDFlib API Features

2.1 Deprecated PDFlib API Methods

Table 2.1 *Deprecated PDFlib API functions/methods and recommended replacements*

deprecated API method	deprecated since	recommended replacement (new API method)
<code>PDF_add_bookmark()</code> <code>PDF_add_bookmark2()</code>	PDFlib 6	<code>PDF_create_bookmark()</code>
<code>PDF_add_launchlink()</code>	PDFlib 6	<code>PDF_create_action()</code> with <code>type=Launch</code> and <code>PDF_create_annotation()</code> with <code>type=Launch</code>
<code>PDF_add_locallink()</code>	PDFlib 6	<code>PDF_create_action()</code> with <code>type=GoTo</code> and <code>PDF_create_annotation()</code> with <code>type=Link</code>
<code>PDF_add_note()</code> <code>PDF_add_note2()</code>	PDFlib 6	<code>PDF_create_annotation()</code> with <code>type=Text</code>
<code>PDF_add_pdflink()</code>	PDFlib 6	<code>PDF_create_action()</code> with <code>type=GoToR</code> and <code>PDF_create_annotation()</code> with <code>type=Link</code>
<code>PDF_add_thumbnail()</code>	PDFlib 9	none; thumbnail creation should be avoided completely since it doesn't have any effect in Acrobat
<code>PDF_add_weblink()</code>	PDFlib 6	<code>PDF_create_action()</code> with <code>type=URI</code> and <code>PDF_create_annotation()</code> with <code>type=Link</code>
<code>PDF_attach_file()</code> <code>PDF_attach_file2()</code>	PDFlib 6	<code>PDF_begin_document()</code> with option <code>attachments</code> or <code>PDF_create_annotation()</code> with <code>type=FileAttachment</code>
<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.o.2	<code>PDF_begin_pattern_ext()</code>
<code>PDF_begin_template()</code>	PDFlib 7	<code>PDF_begin_template_ext()</code>
<code>PDF_boot()</code>	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 keyword <code>major/minor</code>
<code>PDF_get_parameter()</code>	PDFlib 9	<code>PDF_get_option()</code> and <code>PDF_get_string()</code>
<code>PDF_get_pdi_value()</code>	PDFlib 7	<code>PDF_pcos_get_number()</code> <i>pCOS paths corresponding to the keywords of the deprecated function can be found in the pCOS Path Reference</i>
<code>PDF_get_pdi_parameter()</code>	PDFlib 7	<code>PDF_pcos_get_string()</code> <i>pCOS paths corresponding to the keywords of the deprecated function can be found in the pCOS Path Reference</i>
<code>PDF_get_value()</code>	PDFlib 9	<code>PDF_get_option()</code>
<code>PDF_initgraphics()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> with option <code>initgraphicsstate</code>
<code>PDF_open_CCITT()</code>	PDFlib 5	<code>PDF_load_image()</code>
<code>PDF_open_file()</code>	PDFlib 6	<code>PDF_begin_document()</code>
<code>PDF_open_image()</code> <code>PDF_open_image_file()</code>	PDFlib 5	<code>PDF_load_image()</code>

Table 2.1 *Deprecated PDFlib API functions/methods and recommended replacements*

deprecated API method	deprecated since	recommended replacement (new API method)
<code>PDF_open_mem()</code>	PDFlib 6	<code>PDF_begin_document()</code> with empty filename and <code>PDF_get_buffer()</code>
<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>
<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	<code>PDF_set_option()</code> <code>PDF_set_text_option()</code> <code>PDF_set_graphics_option()</code>
<code>PDF_setdash()</code>	PDFlib 9	<code>PDF_set_text_option()</code> and other functions with text appearance options with option <code>dasharray</code> ; <code>PDF_set_graphics_option()</code> and other functions with graphics appearance options with option <code>dasharray</code>
<code>PDF_setdashpattern()</code>	PDFlib 9	<code>PDF_set_graphics_option()</code> with options <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 functions 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_text_option()</code> and other functions with text appearance options with option <code>dasharray</code> ; <code>PDF_set_graphics_option()</code> and other functions 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 functions 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_create_textflow()</code> and <code>PDF_fit_textflow()</code> the options <code>minspacing=100%</code> , <code>maxspacing=10000%</code> , <code>nofitlimit=100%</code> , and <code>shrinklimit=100%</code> achieve similar formatting
<code>PDF_shutdown()</code>	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>
<code>PDF_xshow()</code>	PDFlib 9.0.1	<code>PDF_fit_textline()</code> with option <code>xadvancelist</code>

2.2 Deprecated Options

A list of deprecated options can be found in the document *compatibility.txt* which is included in the PDFlib distribution packages.

2.3 PDFlib History

Table 2.2 PDFlib release history

PDFlib version	release date
PDFlib 4	2001
PDFlib 5	2003
PDFlib 6	2004
PDFlib 7	2006
PDFlib 8	2009
PDFlib 8.1	2011
PDFlib 9	2013
PDFlib 9.1	2016
PDFlib 9.2	2019