

**PDF**

# Introduction

The PDF functions in PHP can create PDF files using the PDFlib library which was initially created by Thomas Merz and is now maintained by [» PDFlib GmbH](#).

The documentation in this section is only meant to be an overview of the available functions in the PDFlib library and should not be considered an exhaustive reference. For the full and detailed explanation of each function, consult the PDFlib Reference Manual which is included in all PDFlib packages distributed by PDFlib GmbH. It provides a very good overview of what PDFlib is capable of doing and contains the most up-to-date documentation of all functions.

For a jump start we urge you to take a look at the programming samples which are contained in all PDFlib distribution packages. These samples demonstrate basic text, vector, and graphics output as well as higher-level functions, such as the PDF import facility (PDI).

All of the functions in PDFlib and the PHP module have identical function names and parameters. Unless configured otherwise, all lengths and coordinates are measured in PostScript points. There are generally 72 PostScript points to an inch, but this depends on the output resolution. Please see the PDFlib Reference Manual included in the PDFlib distribution for a more thorough explanation of the coordinate system used.

With version 6, PDFlib offers an object-oriented API for PHP 5 in addition to the function-oriented API for PHP 4. The main difference is the following:

In PHP 4, first a PDF resource has to be retrieved with a function call like

```
$p = PDF_new().
```

This PDF resource is used as the first parameter in all further function calls, such as in

```
PDF_begin_document($p, "", "").
```

In PHP 5 however, a PDFlib object is created with

```
$p = new PDFlib().
```

This object offers all PDFlib API functions as methods, e.g. as with

```
$p->begin_document("", "").
```

In addition, exceptions have been introduced in PHP 5 which are supported by PDFlib 6 and later as well.

Please see the [examples](#) below for more information.

**Note**

If you're interested in alternative free PDF generators that do not utilize external PDF libraries, see [this related FAQ](#).

# Installing/Configuring

## Requirements

PDFlib Lite is available as open source. However, the PDFlib Lite license allows free use only under certain conditions. PDFlib Lite supports a subset of PDFlib's functionality; please see the PDFlib web site for details. The full version of PDFlib is available for download at » <http://www.pdflib.com/products/pdflib-family/>, but requires that you purchase a license for commercial use.

## Issues with older versions of PDFlib

Any version of PHP 4 after March 9, 2000 does not support versions of PDFlib older than 3.0.

PDFlib 4.0 or greater is supported by PHP 4.3.0 and later.

## Installation

This » [PECL](#) extension is not bundled with PHP. Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: » <http://pecl.php.net/package/pdflib>.

To get these functions to work in PHP < 4.3.9, you have to compile PHP with `--with-pdflib[=DIR]`. DIR is the PDFlib base install directory, defaults to `/usr/local`.

## Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

## Resource Types

[PDF\\_new\(\)](#) creates a new PDFlib object required by most PDF functions.

# Predefined Constants

This extension has no constants defined.

# Examples

## Basic Usage Examples

Most of the functions are fairly easy to use. The most difficult part is probably creating your first PDF document. The following example should help to get you started. It is developed for PHP 4 and creates the file *hello.pdf* with one page. It defines some document info field contents, loads the Helvetica-Bold font and outputs the text "Hello world! (says PHP)".

### Example #1 - Hello World example from PDFlib distribution for PHP 4

```
<?php
$p = PDF_new();

/* open new PDF file; insert a file name to create the PDF on disk */
if (PDF_begin_document($p, "", "") == 0) {
    die("Error: " . PDF_get_errmsg($p));
}

PDF_set_info($p, "Creator", "hello.php");
PDF_set_info($p, "Author", "Rainer Schaaf");
PDF_set_info($p, "Title", "Hello world (PHP)!");

PDF_begin_page_ext($p, 595, 842, "");

$font = PDF_load_font($p, "Helvetica-Bold", "winansi", "");

PDF_setfont($p, $font, 24.0);
PDF_set_text_pos($p, 50, 700);
PDF_show($p, "Hello world!");
PDF_continue_text($p, "(says PHP)");
PDF_end_page_ext($p, "");

PDF_end_document($p, "");

$buf = PDF_get_buffer($p);
$len = strlen($buf);

header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=hello.pdf");
print $buf;

PDF_delete($p);
?>
```

The following example comes with the PDFlib distribution for PHP 5. It uses the new exception handling and object encapsulation features available in PHP 5. It creates the file *hello.pdf* with one page. It defines some document info field contents, loads the

Helvetica-Bold font and outputs the text "Hello world! (says PHP)".

## Example #2 - Hello World example from PDFlib distribution for PHP 5

```
<?php

try {
    $p = new PDFlib();

    /* open new PDF file; insert a file name to create the PDF on disk */
    if ($p->begin_document("", "") == 0) {
        die("Error: " . $p->get_errmsg());
    }

    $p->set_info("Creator", "hello.php");
    $p->set_info("Author", "Rainer Schaaaf");
    $p->set_info("Title", "Hello world (PHP)!");

    $p->begin_page_ext(595, 842, "");

    $font = $p->load_font("Helvetica-Bold", "winansi", "");

    $p->setfont($font, 24.0);
    $p->set_text_pos(50, 700);
    $p->show("Hello world!");
    $p->continue_text("(says PHP)");
    $p->end_page_ext("");

    $p->end_document("");

    $buf = $p->get_buffer();
    $len = strlen($buf);

    header("Content-type: application/pdf");
    header("Content-Length: $len");
    header("Content-Disposition: inline; filename=hello.pdf");
    print $buf;
}
catch (PDFlibException $e) {
    die("PDFlib exception occurred in hello sample:\n" .
        "[" . $e->get_errnum() . "] " . $e->get_apiname() . ": " .
        $e->get_errmsg() . "\n");
}
catch (Exception $e) {
    die($e);
}

$p = 0;
?>
```

# PDF Functions

## Remarks about Deprecated PDFlib Functions

Starting with PHP 4.0.5, the PHP extension for PDFlib is officially supported by PDFlib GmbH. This means that all the functions described in the PDFlib Reference Manual are supported by PHP 4 with exactly the same meaning and the same parameters. However, with PDFlib Version 5.0.4 or higher all parameters have to be specified. For compatibility reasons, this binding for PDFlib still supports most of the deprecated functions, but they should be replaced by their new versions. PDFlib GmbH will not support any problems arising from the use of these deprecated functions. The documentation in this section indicates old functions as "Deprecated" and gives the replacement function to be used instead.



# PDF\_activate\_item

PDF\_activate\_item -- Activate structure element or other content item

## Description

bool **PDF\_activate\_item** ( resource \$pdfdoc, int \$id )

Activates a previously created structure element or other content item. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_add\_annotation

PDF\_add\_annotation -- Add annotation [deprecated]

## Description

This function is deprecated, use [PDF\\_create\\_annotation\(\)](#) with *type=Text* instead.

# PDF\_add\_bookmark

PDF\_add\_bookmark -- Add bookmark for current page [deprecated]

## Description

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_bookmark\(\)](#) instead.

# PDF\_add\_launchlink

PDF\_add\_launchlink -- Add launch annotation for current page [deprecated]

## Description

```
bool PDF_add_launchlink ( resource $pdfdoc, float $llx, float $lly, float $urx, float $
ury, string $filename )
```

Adds a link to a web resource.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_action\(\)](#) with *type=Launch* and [PDF\\_create\\_annotation\(\)](#) with *type=Link* instead.

# PDF\_add\_loccallink

PDF\_add\_loccallink -- Add link annotation for current page [deprecated]

## Description

bool **PDF\_add\_loccallink** ( resource \$pdfdoc, float \$lowerleftx, float \$lowerlefty, float \$upperrightx, float \$upperrighty, int \$page, string \$dest )

Add a link annotation to a target within the current PDF file. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_action\(\)](#) with *type=GoTo* and [PDF\\_create\\_annotation\(\)](#) with *type=Link* instead.

# PDF\_add\_nameddest

PDF\_add\_nameddest -- Create named destination

## Description

bool **PDF\_add\_nameddest** ( resource \$pdfdoc, string \$name, string \$optlist )

Creates a named destination on an arbitrary page in the current document. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_add\_note

PDF\_add\_note -- Set annotation for current page [deprecated]

## Description

bool **PDF\_add\_note** ( resource \$pdfdoc, float \$llx, float \$lly, float \$urx, float \$ury,  
string \$contents, string \$title, string \$icon, int \$open )

Sets an annotation for the current page. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_annotation\(\)](#) with *type=Text* instead.

# PDF\_add\_outline

PDF\_add\_outline -- Add bookmark for current page [deprecated]

## Description

This function is deprecated, use [PDF\\_create\\_bookmark\(\)](#) instead.



# PDF\_add\_pdflink

PDF\_add\_pdflink -- Add file link annotation for current page [deprecated]

## Description

```
bool PDF_add_pdflink ( resource $pdfdoc, float $bottom_left_x, float $bottom_left_y,  
float $up_right_x, float $up_right_y, string $filename, int $page, string $dest )
```

Add a file link annotation to a PDF target. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_action\(\)](#) with *type=GoToR* and [PDF\\_create\\_annotation\(\)](#) with *type=Link* instead.

# PDF\_add\_table\_cell

PDF\_add\_table\_cell -- Add a cell to a new or existing table

## Description

```
int PDF_add_table_cell ( resource $pdfdoc, int $table, int $column, int $row, string $  
text, string $optlist )
```

Adds a cell to a new or existing table.

# PDF\_add\_textflow

PDF\_add\_textflow -- Create Textflow or add text to existing Textflow

## Description

int **PDF\_add\_textflow** ( resource \$pdfdoc, int \$textflow, string \$text, string \$optlist )

Creates a Textflow object, or adds text and explicit options to an existing Textflow.

# PDF\_add\_thumbnail

PDF\_add\_thumbnail -- Add thumbnail for current page

## Description

bool **PDF\_add\_thumbnail** ( resource \$pdfdoc, int \$image )

Adds an existing image as thumbnail for the current page. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_add\_weblink

PDF\_add\_weblink -- Add weblink for current page [deprecated]

## Description

bool **PDF\_add\_weblink** ( resource \$pdfdoc, float \$lowerleftx, float \$lowerlefty, float \$upperrightx, float \$upperrighty, string \$url )

Adds a weblink annotation to a target *url* on the Web. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_action\(\)](#) with *type=URI* and [PDF\\_create\\_annotation\(\)](#) with *type=Link* instead.

# PDF\_arc

PDF\_arc -- Draw a counterclockwise circular arc segment

## Description

bool **PDF\_arc** ( resource \$p, float \$x, float \$y, float \$r, float \$alpha, float \$beta )

Adds a counterclockwise circular arc.

# PDF\_arcn

PDF\_arcn -- Draw a clockwise circular arc segment

## Description

bool **PDF\_arcn** ( resource \$p, float \$x, float \$y, float \$r, float \$alpha, float \$beta )

Except for the drawing direction, this function behaves exactly like [PDF\\_arc\(\)](#).

# PDF\_attach\_file

PDF\_attach\_file -- Add file attachment for current page [deprecated]

## Description

bool **PDF\_attach\_file** ( resource \$pdfdoc, float \$llx, float \$lly, float \$urx, float \$ury, string \$filename, string \$description, string \$author, string \$mimetype, string \$icon )

Adds a file attachment annotation. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_create\\_annotation\(\)](#) with *type=FileAttachment* instead.



# PDF\_begin\_document

PDF\_begin\_document -- Create new PDF file

## Description

```
int PDF_begin_document ( resource $pdfdoc, string $filename, string $optlist )
```

Creates a new PDF file subject to various options.

# PDF\_begin\_font

PDF\_begin\_font -- Start a Type 3 font definition

## Description

bool **PDF\_begin\_font** ( resource \$pdfdoc, string \$filename, float \$a, float \$b, float \$c, float \$d, float \$e, float \$f, string \$optlist )

Starts a Type 3 font definition.

# PDF\_begin\_glyph

PDF\_begin\_glyph -- Start glyph definition for Type 3 font

## Description

bool **PDF\_begin\_glyph** ( resource \$pdfdoc, string \$glyphname, float \$wx, float \$llx, float \$lly, float \$urx, float \$ury )

Starts a glyph definition for a Type 3 font.

# PDF\_begin\_item

PDF\_begin\_item -- Open structure element or other content item

## Description

```
int PDF_begin_item ( resource $pdfdoc, string $tag, string $optlist )
```

Opens a structure element or other content item with attributes supplied as options.

# PDF\_begin\_layer

PDF\_begin\_layer -- Start layer

## Description

bool **PDF\_begin\_layer** ( resource \$pdfdoc, int \$layer )

Starts a layer for subsequent output on the page. Returns **TRUE** on success or **FALSE** on failure.

This function requires PDF 1.5.

# PDF\_begin\_page\_ext

PDF\_begin\_page\_ext -- Start new page

## Description

bool **PDF\_begin\_page\_ext** ( resource \$pdfdoc, float \$width, float \$height, string \$optlist )

Adds a new page to the document, and specifies various options. The parameters *width* and *height* are the dimensions of the new page in points. Returns **TRUE** on success or **FALSE** on failure.

## Common Page Sizes in Points

name	size
A0	2380?3368
A1	1684?2380
A2	1190?1684
A3	842?1190
A4	595?842
A5	421?595
A6	297?421
B5	501?709
letter (8.5"?11")	612?792
legal (8.5"?14")	612?1008
ledger (17"?11")	1224?792
11"?17"	792?1224

# PDF\_begin\_page

PDF\_begin\_page -- Start new page [deprecated]

## Description

bool **PDF\_begin\_page** ( resource \$pdfdoc, float \$width, float \$height )

Adds a new page to the document. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_begin\\_page\\_ext\(\)](#) instead.

# PDF\_begin\_pattern

PDF\_begin\_pattern -- Start pattern definition

## Description

```
int PDF_begin_pattern ( resource $pdfdoc, float $width, float $height, float $xstep, float $ystep, int $painttype )
```

Starts a new pattern definition.



# PDF\_begin\_template\_ext

PDF\_begin\_template\_ext -- Start template definition

## Description

```
int PDF_begin_template_ext ( resource $pdfdoc, float $width, float $height, string $  
optlist )
```

Starts a new template definition.

# PDF\_begin\_template

PDF\_begin\_template -- Start template definition [deprecated]

## Description

int **PDF\_begin\_template** ( resource \$pdfdoc, float \$width, float \$height )

Starts a new template definition.

This function is deprecated since PDFlib version 7, use [PDF\\_begin\\_template\\_ext\(\)](#) instead.

# PDF\_circle

PDF\_circle -- Draw a circle

## Description

bool **PDF\_circle** ( resource \$pdfdoc, float \$x, float \$y, float \$r )

Adds a circle. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_clip

PDF\_clip -- Clip to current path

## Description

bool **PDF\_clip** ( resource \$p )

Uses the current path as clipping path, and terminate the path. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_close\_image

PDF\_close\_image -- Close image

## Description

bool **PDF\_close\_image** ( resource \$p, int \$image )

Closes an *image* retrieved with the [PDF\\_open\\_image\(\)](#) function.

# PDF\_close\_pdi\_page

PDF\_close\_pdi\_page -- Close the page handle

## Description

bool **PDF\_close\_pdi\_page** ( resource *\$p*, int *\$page* )

Closes the page handle, and frees all page-related resources. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_close\_pdi

PDF\_close\_pdi -- Close the input PDF document [deprecated]

## Description

bool **PDF\_close\_pdi** ( resource \$p, int \$doc )

Closes all open page handles, and closes the input PDF document. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 7, use **PDF\_close\_pdi\_document()** instead.

# PDF\_close

PDF\_close -- Close pdf resource [deprecated]

## Description

bool **PDF\_close** ( resource \$p )

Closes the generated PDF file, and frees all document-related resources. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_end\\_document\(\)](#) instead.



# PDF\_closepath\_fill\_stroke

PDF\_closepath\_fill\_stroke -- Close, fill and stroke current path

## Description

bool **PDF\_closepath\_fill\_stroke** ( resource \$p )

Closes the path, fills, and strokes it. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_closepath\_stroke

PDF\_closepath\_stroke -- Close and stroke path

## Description

bool **PDF\_closepath\_stroke** ( resource \$p )

Closes the path, and strokes it. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_closepath

PDF\_closepath -- Close current path

## Description

bool **PDF\_closepath** ( resource \$p )

Closes the current path. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_concat

PDF\_concat -- Concatenate a matrix to the CTM

## Description

bool **PDF\_concat** ( resource \$p, float \$a, float \$b, float \$c, float \$d, float \$e, float \$f )

Concatenates a matrix to the current transformation matrix (CTM). Returns **TRUE** on success or **FALSE** on failure.

# PDF\_continue\_text

PDF\_continue\_text -- Output text in next line

## Description

bool **PDF\_continue\_text** ( resource *\$p*, string *\$text* )

Prints *text* at the next line. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_create\_3dview

PDF\_create\_3dview -- Create 3D view

## Description

int **PDF\_create\_3dview** ( resource \$pdfdoc, string \$username, string \$optlist )

Creates a 3D view.

This function requires PDF 1.6.

# PDF\_create\_action

PDF\_create\_action -- Create action for objects or events

## Description

```
int PDF_create_action ( resource $pdfdoc, string $type, string $optlist )
```

Creates an action which can be applied to various objects and events.

# PDF\_create\_annotation

PDF\_create\_annotation -- Create rectangular annotation

## Description

bool **PDF\_create\_annotation** ( resource \$pdfdoc, float \$llx, float \$lly, float \$urx, float \$ury, string \$type, string \$optlist )

Creates a rectangular annotation on the current page.



# PDF\_create\_bookmark

PDF\_create\_bookmark -- Create bookmark

## Description

```
int PDF_create_bookmark ( resource $pdfdoc, string $text, string $optlist )
```

Creates a bookmark subject to various options.

# PDF\_create\_field

PDF\_create\_field -- Create form field

## Description

bool **PDF\_create\_field** ( resource \$pdfdoc, float \$llx, float \$lly, float \$urx, float \$ury,  
string \$name, string \$type, string \$optlist )

Creates a form field on the current page subject to various options.

# PDF\_create\_fieldgroup

PDF\_create\_fieldgroup -- Create form field group

## Description

bool **PDF\_create\_fieldgroup** ( resource \$pdfdoc, string \$name, string \$optlist )

Creates a form field group subject to various options.

# PDF\_create\_gstate

PDF\_create\_gstate -- Create graphics state object

## Description

int **PDF\_create\_gstate** ( resource \$pdfdoc, string \$optlist )

Creates a graphics state object subject to various options.

# PDF\_create\_pvf

PDF\_create\_pvf -- Create PDFlib virtual file

## Description

```
bool PDF_create_pvf ( resource $pdfdoc, string $filename, string $data, string $optlist  
)
```

Creates a named virtual read-only file from data provided in memory.

# PDF\_create\_textflow

PDF\_create\_textflow -- Create textflow object

## Description

```
int PDF_create_textflow ( resource $pdfdoc, string $text, string $optlist )
```

Preprocesses text for later formatting and creates a textflow object.

# PDF\_curveto

PDF\_curveto -- Draw Bezier curve

## Description

bool **PDF\_curveto** ( resource \$p, float \$x1, float \$y1, float \$x2, float \$y2, float \$x3, float \$y3 )

Draws a Bezier curve from the current point, using 3 more control points. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_define\_layer

PDF\_define\_layer -- Create layer definition

## Description

int **PDF\_define\_layer** ( resource \$pdfdoc, string \$name, string \$optlist )

Creates a new layer definition.

This function requires PDF 1.5.



# PDF\_delete\_pvf

PDF\_delete\_pvf -- Delete PDFlib virtual file

## Description

```
int PDF_delete_pvf ( resource $pdfdoc, string $filename )
```

Deletes a named virtual file and frees its data structures (but not the contents).

# PDF\_delete\_table

PDF\_delete\_table -- Delete table object

## Description

bool **PDF\_delete\_table** ( resource \$pdfdoc, int \$table, string \$optlist )

Deletes a table and all associated data structures.

# PDF\_delete\_textflow

PDF\_delete\_textflow -- Delete textflow object

## Description

bool **PDF\_delete\_textflow** ( resource \$pdfdoc, int \$textflow )

Deletes a textflow and the associated data structures.

# PDF\_delete

PDF\_delete -- Delete PDFlib object

## Description

bool **PDF\_delete** ( resource \$pdfdoc )

Deletes a PDFlib object, and frees all internal resources. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_encoding\_set\_char

PDF\_encoding\_set\_char -- Add glyph name and/or Unicode value

## Description

```
bool PDF_encoding_set_char ( resource $pdfdoc, string $encoding, int $slot, string $  
glyphname, int $uv )
```

Adds a glyph name and/or Unicode value to a custom encoding.

# PDF\_end\_document

PDF\_end\_document -- Close PDF file

## Description

bool **PDF\_end\_document** ( resource \$pdfdoc, string \$optlist )

Closes the generated PDF file and applies various options.

# PDF\_end\_font

PDF\_end\_font -- Terminate Type 3 font definition

## Description

bool **PDF\_end\_font** ( resource \$pdfdoc )

Terminates a Type 3 font definition.

# PDF\_end\_glyph

PDF\_end\_glyph -- Terminate glyph definition for Type 3 font

## Description

bool **PDF\_end\_glyph** ( resource \$pdfdoc )

Terminates a glyph definition for a Type 3 font.



# PDF\_end\_item

PDF\_end\_item -- Close structure element or other content item

## Description

bool **PDF\_end\_item** ( resource \$pdfdoc, int \$id )

Closes a structure element or other content item.

# PDF\_end\_layer

PDF\_end\_layer -- Deactivate all active layers

## Description

bool **PDF\_end\_layer** ( resource \$pdfdoc )

Deactivates all active layers. Returns **TRUE** on success or **FALSE** on failure.

This function requires PDF 1.5.

# PDF\_end\_page\_ext

PDF\_end\_page\_ext -- Finish page

## Description

bool **PDF\_end\_page\_ext** ( resource \$pdfdoc, string \$optlist )

Finishes a page, and applies various options. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_end\_page

PDF\_end\_page -- Finish page

## Description

bool **PDF\_end\_page** ( resource \$p )

Finishes the page. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_end\_pattern

PDF\_end\_pattern -- Finish pattern

## Description

bool **PDF\_end\_pattern** ( resource \$p )

Finishes the pattern definition. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_end\_template

PDF\_end\_template -- Finish template

## Description

bool **PDF\_end\_template** ( resource \$p )

Finishes a template definition. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_endpath

PDF\_endpath -- End current path

## Description

bool **PDF\_endpath** ( resource *\$p* )

Ends the current path without filling or stroking it.

# PDF\_fill\_imageblock

PDF\_fill\_imageblock -- Fill image block with variable data

## Description

```
int PDF_fill_imageblock ( resource $pdfdoc, int $page, string $blockname, int $image,  
string $optlist )
```

Fills an image block with variable data according to its properties.

This function is only available in the PDFlib Personalization Server (PPS).



# PDF\_fill\_pdfblock

PDF\_fill\_pdfblock -- Fill PDF block with variable data

## Description

```
int PDF_fill_pdfblock ( resource $pdfdoc, int $page, string $blockname, int $contents,  
string $optlist )
```

Fills a PDF block with variable data according to its properties.

This function is only available in the PDFlib Personalization Server (PPS).

# PDF\_fill\_stroke

PDF\_fill\_stroke -- Fill and stroke path

## Description

bool **PDF\_fill\_stroke** ( resource \$p )

Fills and strokes the current path with the current fill and stroke color. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_fill\_textblock

PDF\_fill\_textblock -- Fill text block with variable data

## Description

```
int PDF_fill_textblock ( resource $pdfdoc, int $page, string $blockname, string $text,  
string $optlist )
```

Fills a text block with variable data according to its properties.

This function is only available in the PDFlib Personalization Server (PPS).

# PDF\_fill

PDF\_fill -- Fill current path

## Description

bool **PDF\_fill** ( resource \$p )

Fills the interior of the current path with the current fill color. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_findfont

PDF\_findfont -- Prepare font for later use [deprecated]

## Description

```
int PDF_findfont ( resource $p, string $fontname, string $encoding, int $embed )
```

Search for a font and prepare it for later use with [PDF\\_setfont\(\)](#). The metrics will be loaded, and if *embed* is nonzero, the font file will be checked, but not yet used. *encoding* is one of *builtin*, *macroman*, *winansi*, *host*, a user-defined encoding name or the name of a CMap. Parameter *embed* is optional before PHP 4.3.5 or with PDFlib less than 5.

This function is deprecated since PDFlib version 5, use [PDF\\_load\\_font\(\)](#) instead.

# PDF\_fit\_image

PDF\_fit\_image -- Place image or template

## Description

bool **PDF\_fit\_image** ( resource \$pdfdoc, int \$image, float \$x, float \$y, string \$optlist )

Places an image or template on the page, subject to various options. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_fit\_pdi\_page

PDF\_fit\_pdi\_page -- Place imported PDF page

## Description

bool **PDF\_fit\_pdi\_page** ( resource \$pdfdoc, int \$page, float \$x, float \$y, string \$optlist )

Places an imported PDF page on the page, subject to various options. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_fit\_table

PDF\_fit\_table -- Place table on page

## Description

string **PDF\_fit\_table** ( resource \$pdfdoc, int \$table, float \$llx, float \$lly, float \$urx, float \$ury, string \$optlist )

Places a table on the page fully or partially.



# PDF\_fit\_textflow

PDF\_fit\_textflow -- Format textflow in rectangular area

## Description

string **PDF\_fit\_textflow** ( resource \$pdfdoc, int \$textflow, float \$llx, float \$lly, float \$  
urx, float \$ury, string \$optlist )

Formats the next portion of a textflow into a rectangular area.

# PDF\_fit\_textline

PDF\_fit\_textline -- Place single line of text

## Description

```
bool PDF_fit_textline ( resource $pdfdoc, string $text, float $x, float $y, string $optlist  
)
```

Places a single line of text on the page, subject to various options. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_get\_apiname

PDF\_get\_apiname -- Get name of unsuccessful API function

## Description

string **PDF\_get\_apiname** ( resource \$pdfdoc )

Gets the name of the API function which threw the last exception or failed.

# PDF\_get\_buffer

PDF\_get\_buffer -- Get PDF output buffer

## Description

string **PDF\_get\_buffer** ( resource \$p )

Fetches the buffer containing the generated PDF data.

# PDF\_get\_errmsg

PDF\_get\_errmsg -- Get error text

## Description

string **PDF\_get\_errmsg** ( resource `$pdfdoc` )

Gets the text of the last thrown exception or the reason for a failed function call.

# PDF\_get\_errnum

PDF\_get\_errnum -- Get error number

## Description

int **PDF\_get\_errnum** ( resource \$pdfdoc )

Gets the number of the last thrown exception or the reason for a failed function call.

# PDF\_get\_font

PDF\_get\_font -- Get font [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_get\\_value\(\)](#) with the parameter *font* instead.

# PDF\_get\_fontname

PDF\_get\_fontname -- Get font name [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_get\\_parameter\(\)](#) with the parameter *fontname* instead.



# PDF\_get\_fontsize

PDF\_get\_fontsize -- Font handling [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_get\\_value\(\)](#) with the parameter *fontsize* instead.

# PDF\_get\_image\_height

PDF\_get\_image\_height -- Get image height [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_get\\_value\(\)](#) with the parameter *imageheight* instead.

# PDF\_get\_image\_width

PDF\_get\_image\_width -- Get image width [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_get\\_value\(\)](#) with the parameter *imagewidth* instead.

# PDF\_get\_majorversion

PDF\_get\_majorversion -- Get major version number [deprecated]

## Description

int **PDF\_get\_majorversion** ( void )

This function is deprecated since PDFlib version 5, use [PDF\\_get\\_value\(\)](#) with the parameter *major* instead.

# PDF\_get\_minorversion

PDF\_get\_minorversion -- Get minor version number [deprecated]

## Description

int **PDF\_get\_minorversion** ( void )

Returns the minor version number of the PDFlib version.

This function is deprecated since PDFlib version 5, use [PDF\\_get\\_value\(\)](#) with the parameter *minor* instead.

# PDF\_get\_parameter

PDF\_get\_parameter -- Get string parameter

## Description

string **PDF\_get\_parameter** ( resource \$p, string \$key, float \$modifier )

Gets the contents of some PDFlib parameter with string type.

# PDF\_get\_pdi\_parameter

PDF\_get\_pdi\_parameter -- Get PDI string parameter [deprecated]

## Description

string **PDF\_get\_pdi\_parameter** ( resource \$p, string \$key, int \$doc, int \$page, int \$reserved )

Gets the contents of a PDI document parameter with string type.

This function is deprecated since PDFlib version 7, use [PDF\\_pcos\\_get\\_string\(\)](#) instead.

# PDF\_get\_pdi\_value

PDF\_get\_pdi\_value -- Get PDI numerical parameter [deprecated]

## Description

float **PDF\_get\_pdi\_value** ( resource \$p, string \$key, int \$doc, int \$page, int \$reserved )

Gets the contents of a PDI document parameter with numerical type.

This function is deprecated since PDFlib version 7, use [PDF\\_pcos\\_get\\_number\(\)](#) instead.



# PDF\_get\_value

PDF\_get\_value -- Get numerical parameter

## Description

float **PDF\_get\_value** ( resource \$p, string \$key, float \$modifier )

Gets the value of some PDFlib parameter with numerical type.

# PDF\_info\_font

PDF\_info\_font -- Query detailed information about a loaded font

## Description

float **PDF\_info-font** ( resource \$pdfdoc, int \$font, string \$keyword, string \$optlist )

Queries detailed information about a loaded font.

# PDF\_info\_matchbox

PDF\_info\_matchbox -- Query matchbox information

## Description

```
float PDF_info_matchbox ( resource $pdfdoc, string $boxname, int $num, string $keyword  
)
```

Queries information about a matchbox on the current page.

# PDF\_info\_table

PDF\_info\_table -- Retrieve table information

## Description

float **PDF\_info\_table** ( resource \$pdfdoc, int \$table, string \$keyword )

Retrieves table information related to the most recently placed table instance.

# PDF\_info\_textflow

PDF\_info\_textflow -- Query textflow state

## Description

float **PDF\_info\_textflow** ( resource \$pdfdoc, int \$textflow, string \$keyword )

Queries the current state of a textflow.

# PDF\_info\_textline

PDF\_info\_textline -- Perform textline formatting and query metrics

## Description

float **PDF\_info\_textline** ( resource \$pdfdoc, string \$text, string \$keyword, string \$optlist )

Performs textline formatting and queries the resulting metrics.

# PDF\_initgraphics

PDF\_initgraphics -- Reset graphic state

## Description

bool **PDF\_initgraphics** ( resource \$p )

Reset all color and graphics state parameters to their defaults. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_lineto

PDF\_lineto -- Draw a line

## Description

bool **PDF\_lineto** ( resource  $\$p$ , float  $\$x$ , float  $\$y$  )

Draws a line from the current point to another point. Returns **TRUE** on success or **FALSE** on failure.



# PDF\_load\_3ddata

PDF\_load\_3ddata -- Load 3D model

## Description

int **PDF\_load\_3ddata** ( resource \$pdfdoc, string \$filename, string \$optlist )

Loads a 3D model from a disk-based or virtual file.

This function requires PDF 1.6.

# PDF\_load\_font

PDF\_load\_font -- Search and prepare font

## Description

```
int PDF_load_font ( resource $pdfdoc, string $fontname, string $encoding, string $  
optlist )
```

Searches for a font and prepares it for later use.

# PDF\_load\_iccprofile

PDF\_load\_iccprofile -- Search and prepare ICC profile

## Description

int **PDF\_load\_iccprofile** ( resource \$pdfdoc, string \$profilename, string \$optlist )

Searches for an ICC profile, and prepares it for later use.

# PDF\_load\_image

PDF\_load\_image -- Open image file

## Description

```
int PDF_load_image ( resource $pdfdoc, string $imagetype, string $filename, string $  
optlist )
```

Opens a disk-based or virtual image file subject to various options.

# PDF\_makespotcolor

PDF\_makespotcolor -- Make spot color

## Description

int **PDF\_makespotcolor** ( resource \$p, string \$spotname )

Finds a built-in spot color name, or makes a named spot color from the current fill color.  
Returns **TRUE** on success or **FALSE** on failure.

# PDF\_moveto

PDF\_moveto -- Set current point

## Description

bool **PDF\_moveto** ( resource *\$p*, float *\$x*, float *\$y* )

Sets the current point for graphics output. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_new

PDF\_new -- Create PDFlib object

## Description

resource **PDF\_new** ( )

Creates a new PDFlib object with default settings.

# PDF\_open\_ccitt

PDF\_open\_ccitt -- Open raw CCITT image [deprecated]

## Description

```
int PDF_open_ccitt ( resource $pdfdoc, string $filename, int $width, int $height, int $  
BitReverse, int $k, int $BlackIs1 )
```

Opens a raw CCITT image.

This function is deprecated since PDFlib version 5, use [PDF\\_load\\_image\(\)](#) instead.



# PDF\_open\_file

PDF\_open\_file -- Create PDF file [deprecated]

## Description

bool **PDF\_open\_file** ( resource \$p, string \$filename )

Creates a new PDF file using the supplied file name. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use [PDF\\_begin\\_document\(\)](#) instead.

# PDF\_open\_gif

PDF\_open\_gif -- Open GIF image [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_load\\_image\(\)](#) instead.

# PDF\_open\_image\_file

PDF\_open\_image\_file -- Read image from file [deprecated]

## Description

```
int PDF_open_image_file ( resource $p, string $imagetype, string $filename, string $  
stringparam, int $intparam )
```

Opens an image file.

This function is deprecated since PDFlib version 5, use [PDF\\_load\\_image\(\)](#) with the `colorize`, `ignoremask`, `invert`, `mask`, `masked`, and `page` options instead.

# PDF\_open\_image

PDF\_open\_image -- Use image data [deprecated]

## Description

int **PDF\_open\_image** ( resource \$p, string \$imagetype, string \$source, string \$data, int \$length, int \$width, int \$height, int \$components, int \$bpc, string \$params )

Uses image data from a variety of data sources.

This function is deprecated since PDFlib version 5, use virtual files and [PDF\\_load\\_image\(\)](#) instead.

# PDF\_open\_jpeg

PDF\_open\_jpeg -- Open JPEG image [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_load\\_image\(\)](#) instead.

# PDF\_open\_memory\_image

PDF\_open\_memory\_image -- Open image created with PHP's image functions [not supported]

## Description

```
int PDF_open_memory_image ( resource $p, resource $image )
```

This function is not supported by PDFlib GmbH.

# PDF\_open\_pdi\_page

PDF\_open\_pdi\_page -- Prepare a page

## Description

int **PDF\_open\_pdi\_page** ( resource \$p, int \$doc, int \$pagenumber, string \$optlist )

Prepares a page for later use with [PDF\\_fit\\_pdi\\_page\(\)](#).

# PDF\_open\_pdi

PDF\_open\_pdi -- Open PDF file [deprecated]

## Description

int **PDF\_open\_pdi** ( resource \$pdfdoc, string \$filename, string \$optlist, int \$len )

Opens a disk-based or virtual PDF document and prepares it for later use.

This function is deprecated since PDFlib version 7, use **PDF\_open\_pdi\_document()** instead.



# PDF\_open\_tiff

PDF\_open\_tiff -- Open TIFF image [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_load\\_image\(\)](#) instead.

# PDF\_pcos\_get\_number

PDF\_pcos\_get\_number -- Get value of pCOS path with type number or boolean

## Description

float **PDF\_pcos\_get\_number** ( resource \$p, int \$doc, string \$path )

Gets the value of a pCOS path with type number or boolean.

# PDF\_pcos\_get\_stream

PDF\_pcos\_get\_stream -- Get contents of pCOS path with type stream, fstream, or string

## Description

string **PDF\_pcos\_get\_stream** ( resource \$p, int \$doc, string \$optlist, string \$path )

Gets the contents of a pCOS path with type stream, fstream, or string.

# PDF\_pcos\_get\_string

PDF\_pcos\_get\_string -- Get value of pCOS path with type name, string, or boolean

## Description

string **PDF\_pcos\_get\_string** ( resource *\$p*, int *\$doc*, string *\$path* )

Gets the value of a pCOS path with type name, string, or boolean.

# PDF\_place\_image

PDF\_place\_image -- Place image on the page [deprecated]

## Description

bool **PDF\_place\_image** ( resource \$pdfdoc, int \$image, float \$x, float \$y, float \$scale )

Places an image and scales it. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 5, use [PDF\\_fit\\_image\(\)](#) instead.

# PDF\_place\_pdi\_page

PDF\_place\_pdi\_page -- Place PDF page [deprecated]

## Description

bool **PDF\_place\_pdi\_page** ( resource \$pdfdoc, int \$page, float \$x, float \$y, float \$sx, float \$sy )

Places a PDF page and scales it. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 5, use [PDF\\_fit\\_pdi\\_page\(\)](#) instead.

# PDF\_process\_pdi

PDF\_process\_pdi -- Process imported PDF document

## Description

```
int PDF_process_pdi ( resource $pdfdoc, int $doc, int $page, string $optlist )
```

Processes certain elements of an imported PDF document.

# PDF\_rect

PDF\_rect -- Draw rectangle

## Description

bool **PDF\_rect** ( resource \$p, float \$x, float \$y, float \$width, float \$height )

Draws a rectangle. Returns **TRUE** on success or **FALSE** on failure.



# PDF\_restore

PDF\_restore -- Restore graphics state

## Description

bool **PDF\_restore** ( resource \$p )

Restores the most recently saved graphics state. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_resume\_page

PDF\_resume\_page -- Resume page

## Description

bool **PDF\_resume\_page** ( resource \$pdfdoc, string \$optlist )

Resumes a page to add more content to it.

# PDF\_rotate

PDF\_rotate -- Rotate coordinate system

## Description

bool **PDF\_rotate** ( resource *\$p*, float *\$phi* )

Rotates the coordinate system. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_save

PDF\_save -- Save graphics state

## Description

bool **PDF\_save** ( resource \$p )

Saves the current graphics state. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_scale

PDF\_scale -- Scale coordinate system

## Description

bool **PDF\_scale** ( resource *\$p*, float *\$sx*, float *\$sy* )

Scales the coordinate system. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_set\_border\_color

PDF\_set\_border\_color -- Set border color of annotations [deprecated]

## Description

bool **PDF\_set\_border\_color** ( resource \$p, float \$red, float \$green, float \$blue )

Sets the border color for all kinds of annotations. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use the option *annotcolor* in [PDF\\_create\\_annotation\(\)](#) instead.

# PDF\_set\_border\_dash

PDF\_set\_border\_dash -- Set border dash style of annotations [deprecated]

## Description

bool **PDF\_set\_border\_dash** ( resource \$pdfdoc, float \$black, float \$white )

Sets the border dash style for all kinds of annotations. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use the option *dasharray* in [PDF\\_create\\_annotation\(\)](#) instead.

# PDF\_set\_border\_style

PDF\_set\_border\_style -- Set border style of annotations [deprecated]

## Description

bool **PDF\_set\_border\_style** ( resource \$pdfdoc, string \$style, float \$width )

Sets the border style for all kinds of annotations. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 6, use the options *borderstyle* and *linewidth* in [PDF\\_create\\_annotation\(\)](#) instead.



# PDF\_set\_char\_spacing

PDF\_set\_char\_spacing -- Set character spacing [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with parameter *charspacing* instead.

# PDF\_set\_duration

PDF\_set\_duration -- Set duration between pages [deprecated]

## Description

This function is deprecated since PDFlib version 3, use the *duration* option in [PDF\\_begin\\_page\\_ext\(\)](#) or [PDF\\_end\\_page\\_ext\(\)](#) instead.

# PDF\_set\_gstate

PDF\_set\_gstate -- Activate graphics state object

## Description

bool **PDF\_set\_gstate** ( resource \$pdfdoc, int \$gstate )

Activates a graphics state object.

# PDF\_set\_horiz\_scaling

PDF\_set\_horiz\_scaling -- Set horizontal text scaling [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with parameter *horizscaling* instead.

# PDF\_set\_info\_author

PDF\_set\_info\_author -- Fill the author document info field [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_info\(\)](#) instead.

# PDF\_set\_info\_creator

PDF\_set\_info\_creator -- Fill the creator document info field [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_info\(\)](#) instead.

# PDF\_set\_info\_keywords

PDF\_set\_info\_keywords -- Fill the keywords document info field [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_info\(\)](#) instead.

# PDF\_set\_info\_subject

PDF\_set\_info\_subject -- Fill the subject document info field [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_info\(\)](#) instead.



# PDF\_set\_info\_title

PDF\_set\_info\_title -- Fill the title document info field [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_info\(\)](#) instead.

# PDF\_set\_info

PDF\_set\_info -- Fill document info field

## Description

bool **PDF\_set\_info** ( resource *\$p*, string *\$key*, string *\$value* )

Fill document information field *key* with *value*. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_set\_layer\_dependency

PDF\_set\_layer\_dependency -- Define relationships among layers

## Description

bool **PDF\_set\_layer\_dependency** ( resource \$pdfdoc, string \$type, string \$optlist )

Defines hierarchical and group relationships among layers. Returns **TRUE** on success or **FALSE** on failure.

This function requires PDF 1.5.

# PDF\_set\_leading

PDF\_set\_leading -- Set distance between text lines [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with the parameter *leading* instead.

# PDF\_set\_parameter

PDF\_set\_parameter -- Set string parameter

## Description

bool **PDF\_set\_parameter** ( resource \$p, string \$key, string \$value )

Sets some PDFlib parameter with string type. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_set\_text\_matrix

PDF\_set\_text\_matrix -- Set text matrix [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_scale\(\)](#), [PDF\\_translate\(\)](#), [PDF\\_rotate\(\)](#), or [PDF\\_skew\(\)](#) instead.

# PDF\_set\_text\_pos

PDF\_set\_text\_pos -- Set text position

## Description

bool **PDF\_set\_text\_pos** ( resource *\$p*, float *\$x*, float *\$y* )

Sets the position for text output on the page. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_set\_text\_rendering

PDF\_set\_text\_rendering -- Determine text rendering [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with the *textrendering* parameter instead.



# PDF\_set\_text\_rise

PDF\_set\_text\_rise -- Set text rise [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with the *textrise* parameter instead.

# PDF\_set\_value

PDF\_set\_value -- Set numerical parameter

## Description

bool **PDF\_set\_value** ( resource \$p, string \$key, float \$value )

Sets the value of some PDFlib parameter with numerical type. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_set\_word\_spacing

PDF\_set\_word\_spacing -- Set spacing between words [deprecated]

## Description

This function is deprecated since PDFlib version 3, use [PDF\\_set\\_value\(\)](#) with the *wordspacing* parameter instead.

# PDF\_setcolor

PDF\_setcolor -- Set fill and stroke color

## Description

bool **PDF\_setcolor** ( resource \$p, string \$fstype, string \$colorspace, float \$c1, float \$c2, float \$c3, float \$c4 )

Sets the current color space and color. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setdash

PDF\_setdash -- Set simple dash pattern

## Description

bool **PDF\_setdash** ( resource *\$pdfdoc*, float *\$b*, float *\$w* )

Sets the current dash pattern to *b* black and *w* white units. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setdashpattern

PDF\_setdashpattern -- Set dash pattern

## Description

bool **PDF\_setdashpattern** ( resource \$pdfdoc, string \$optlist )

Sets a dash pattern defined by an option list. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setflat

PDF\_setflat -- Set flatness

## Description

bool **PDF\_setflat** ( resource \$pdfdoc, float \$flatness )

Sets the flatness parameter. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setfont

PDF\_setfont -- Set font

## Description

bool **PDF\_setfont** ( resource *\$pdfdoc*, int *\$font*, float *\$fontsize* )

Sets the current font in the specified *size*, using a *font* handle returned by [PDF\\_load\\_font\(\)](#). Returns **TRUE** on success or **FALSE** on failure.



# PDF\_setgray\_fill

PDF\_setgray\_fill -- Set fill color to gray [deprecated]

## Description

bool **PDF\_setgray\_fill** ( resource *\$p*, float *\$g* )

Sets the current fill color to a gray value between 0 and 1 inclusive. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_setgray\_stroke

PDF\_setgray\_stroke -- Set stroke color to gray [deprecated]

## Description

bool **PDF\_setgray\_stroke** ( resource *\$p*, float *\$g* )

Sets the current stroke color to a gray value between 0 and 1 inclusive. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_setgray

PDF\_setgray -- Set color to gray [deprecated]

## Description

bool **PDF\_setgray** ( resource *\$p*, float *\$g* )

Sets the current fill and stroke color to a gray value between 0 and 1 inclusive. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_setlinecap

PDF\_setlinecap -- Set linecap parameter

## Description

bool **PDF\_setlinecap** ( resource \$p, int \$linecap )

Sets the *linecap* parameter to control the shape at the end of a path with respect to stroking.

# PDF\_setlinejoin

PDF\_setlinejoin -- Set linejoin parameter

## Description

bool **PDF\_setlinejoin** ( resource *\$p*, int *\$value* )

Sets the *linejoin* parameter to specify the shape at the corners of paths that are stroked.  
Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setlinewidth

PDF\_setlinewidth -- Set line width

## Description

bool **PDF\_setlinewidth** ( resource *\$p*, float *\$width* )

Sets the current line width. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setmatrix

PDF\_setmatrix -- Set current transformation matrix

## Description

bool **PDF\_setmatrix** ( resource \$p, float \$a, float \$b, float \$c, float \$d, float \$e, float \$f )

Explicitly sets the current transformation matrix. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_setmiterlimit

PDF\_setmiterlimit -- Set miter limit

## Description

bool **PDF\_setmiterlimit** ( resource \$pdfdoc, float \$miter )

Sets the miter limit. Returns **TRUE** on success or **FALSE** on failure.



# PDF\_setpolydash

PDF\_setpolydash -- Set complicated dash pattern [deprecated]

## Description

This function is deprecated since PDFlib version 5, use [PDF\\_setdashpattern\(\)](#) instead.

# PDF\_setrgbcolor\_fill

PDF\_setrgbcolor\_fill -- Set fill rgb color values [deprecated]

## Description

bool **PDF\_setrgbcolor\_fill** ( resource \$p, float \$red, float \$green, float \$blue )

Sets the current fill color to the supplied RGB values. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_setrgbcolor\_stroke

PDF\_setrgbcolor\_stroke -- Set stroke rgb color values [deprecated]

## Description

bool **PDF\_setrgbcolor\_stroke** ( resource \$p, float \$red, float \$green, float \$blue )

Sets the current stroke color to the supplied RGB values. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_setrgbcolor

PDF\_setrgbcolor -- Set fill and stroke rgb color values [deprecated]

## Description

bool **PDF\_setrgbcolor** ( resource \$p, float \$red, float \$green, float \$blue )

Sets the current fill and stroke color to the supplied RGB values. Returns **TRUE** on success or **FALSE** on failure.

This function is deprecated since PDFlib version 4, use [PDF\\_setcolor\(\)](#) instead.

# PDF\_shading\_pattern

PDF\_shading\_pattern -- Define shading pattern

## Description

int **PDF\_shading\_pattern** ( resource \$pdfdoc, int \$shading, string \$optlist )

Defines a shading pattern using a shading object.

This function requires PDF 1.4 or above.

# PDF\_shading

PDF\_shading -- Define blend

## Description

int **PDF\_shading** ( resource \$pdfdoc, string \$shtype, float \$x0, float \$y0, float \$x1, float \$y1, float \$c1, float \$c2, float \$c3, float \$c4, string \$optlist )

Defines a blend from the current fill color to another color.

This function requires PDF 1.4 or above.

# PDF\_shfill

PDF\_shfill -- Fill area with shading

## Description

bool **PDF\_shfill** ( resource \$pdfdoc, int \$shading )

Fills an area with a shading, based on a shading object.

This function requires PDF 1.4 or above.

# PDF\_show\_boxed

PDF\_show\_boxed -- Output text in a box [deprecated]

## Description

```
int PDF_show_boxed ( resource $p, string $text, float $left, float $top, float $width,  
float $height, string $mode, string $feature )
```

This function is deprecated since PDFlib version 6, use [PDF\\_fit\\_textline\(\)](#) for single lines, or the **PDF\_\*\_textflow()** functions for multi-line formatting instead.



# PDF\_show\_xy

PDF\_show\_xy -- Output text at given position

## Description

bool **PDF\_show\_xy** ( resource *\$p*, string *\$text*, float *\$x*, float *\$y* )

Prints *text* in the current font. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_show

PDF\_show -- Output text at current position

## Description

bool **PDF\_show** ( resource *\$pdfdoc*, string *\$text* )

Prints *text* in the current font and size at the current position. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_skew

PDF\_skew -- Skew the coordinate system

## Description

bool **PDF\_skew** ( resource *\$p*, float *\$alpha*, float *\$beta* )

Skews the coordinate system in x and y direction by *alpha* and *beta* degrees. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_stringwidth

PDF\_stringwidth -- Return width of text

## Description

float **PDF\_stringwidth** ( resource *\$p*, string *\$text*, int *\$font*, float *\$fontsize* )

Returns the width of *text* in an arbitrary font.

# PDF\_stroke

PDF\_stroke -- Stroke path

## Description

bool **PDF\_stroke** ( resource \$p )

Strokes the path with the current color and line width, and clear it. Returns **TRUE** on success or **FALSE** on failure.

# PDF\_suspend\_page

PDF\_suspend\_page -- Suspend page

## Description

bool **PDF\_suspend\_page** ( resource \$pdfdoc, string \$optlist )

Suspends the current page so that it can later be resumed with [PDF\\_resume\\_page\(\)](#).

# PDF\_translate

PDF\_translate -- Set origin of coordinate system

## Description

bool **PDF\_translate** ( resource \$p, float \$tx, float \$ty )

Translates the origin of the coordinate system.

# PDF\_utf16\_to\_utf8

PDF\_utf16\_to\_utf8 -- Convert string from UTF-16 to UTF-8

## Description

string **PDF\_utf16\_to\_utf8** ( resource \$pdfdoc, string \$utf16string )

Converts a string from UTF-16 format to UTF-8.



# PDF\_utf32\_to\_utf16

PDF\_utf32\_to\_utf16 -- Convert string from UTF-32 to UTF-16

## Description

string **PDF\_utf32\_to\_utf16** ( resource \$pdfdoc, string \$utf32string, string \$ordering )

Converts a string from UTF-32 format to UTF-16.

# PDF\_utf8\_to\_utf16

PDF\_utf8\_to\_utf16 -- Convert string from UTF-8 to UTF-16

## Description

string **PDF\_utf8\_to\_utf16** ( resource \$pdfdoc, string \$utf8string, string \$ordering )

Converts a string from UTF-8 format to UTF-16.