

How to use PDFlib products with PHP

Last change: April 15, 2008

Latest PDFlib version covered in this document: 7.0.3

Latest version of this document available at:

www.pdflib.com/developer/technical-documentation

Contact:

PDFlib GmbH

Franziska-Bilek-Weg 9

80339 München, Germany

phone +49 • 89 • 452 33 84-0

support@pdflib.com

www.pdflib.com

1 Scope of this Document

This document explains various possibilities for successfully deploying PDFlib as a PHP extension. The generic term PDFlib is used to designate one of the following distinct products:

- ▶ PDFlib Lite, the open-source subset of PDFlib
- ▶ The commercial PDFlib product
- ▶ PDFlib+PDI, a commercial superset of PDFlib which also contains the PDF Import Library (PDI)
- ▶ PDFlib Personalization Server (PPS), a superset of PDFlib+PDI with advanced block filling features for personalizing PDF documents.

Notes for PDFlib TET (Text Extraction Toolkit), PDFlib PLOP (Linearization, Optimization, Protection), and PDFlib pCOS (PDF Information Retrieval Tool) are included where applicable.

The methods for deploying any of these products as a PHP extension are the same in all cases. However, only one product of the PDFlib family can be used at a time. Similarly, multiple versions of these products cannot be deployed at the same time. Different products can coexist within one PHP installation, however. Note that the evaluation versions of commercial PDFlib products will be fully functional, but will display a demo stamp across all generated PDF pages unless a valid license key is applied.

This document applies to the following software versions:

- ▶ PDFlib 7.0.3 and PDFlib 6.0.4
- ▶ TET 2.3, PLOP 3.0, and pCOS 2.0
- ▶ PHP 4.3.x, 4.4.x, 5.0.x, 5.1.x, and 5.2.x

Where applicable, version-specific information is provided separately.

2 Classify your System – and yourself

2.1 Platforms where PHP reliably supports DSOs

Loadable PHP extension modules implemented as DSOs (dynamic shared objects, also called dynamic link library DLL) are the recommended method of using PDFlib with PHP. PHP supports dynamic loading of extensions from DSOs on the following platforms (only platforms supported by PDFlib GmbH are mentioned here):

- ▶ Microsoft Windows
- ▶ Mac OS X (see below for details)
- ▶ Linux on x86 and ia64
- ▶ Linux on zSeries
- ▶ FreeBSD 5/6 on x86
- ▶ Sun Solaris 7/8/9/10 on Sparc
- ▶ Sun Solaris 10 on x86
- ▶ HP-UX 11

The PDFlib 7.0.3 and 6.0.4 distribution packages shipped by PDFlib GmbH contain PDFlib DSOs for a number of PHP versions. These are grouped into several directories as follows (not all PHP versions are supported on all platforms, though):

- ▶ *bind/php4/php-430* for PHP 4.3.0 – 4.3.11 and 4.4.0 – 4.4.8
- ▶ *bind/php5/php-500* for PHP 5.0.0 – 5.0.2
- ▶ *bind/php5/php-503* for PHP 5.0.3 – 5.0.5
- ▶ *bind/php5/php-510* for PHP 5.1.0 – 5.1.6
- ▶ *bind/php5/php-520* for PHP 5.2.0 – 5.2.5

Depending on the compatibility properties of the PHP distribution PDFlib may also work with newer versions of PHP, but we have only tested the combinations above.

PHP on Mac OS X. Apple's PHP version which is bundled with Mac OS X does not work with PDFlib DSOs. To use PHP with PDFlib on Mac OS X you need third-party PHP packages such as MAMP, XAMP for Mac, or Marc Liyanage's version from www.entropy.ch. Mac OS X 10.5 (Leopard) adds new complications. As described in developer.apple.com/releasenotes/CoreFoundation/CoreFoundation.html

it is no longer possible to use CoreFoundation functions after a call to `fork()` without `exec()`. However, CoreFoundation functions are required for PDFlib's host font feature, and the critical sequence above is used in the combination of Apache and PHP. This may trigger the following error message in the Apache log (and can even crash the Apache process):

```
The process has forked and you cannot use this CoreFoundation functionality safely. You MUST exec(). Break on _THE_PROCESS_HAS_FORKED_AND_YOU_CANNOT_USE_THIS_COREFOUNDATION_FUNCTIONALITY___YOU_MUST_EXEC__() to debug.
```

In order to avoid this problem you can run PHP as a CGI on Apache, or disable the host font feature in PDFlib using the following call:

```
PDF_set_parameter($p, "debug", "h");
```

Using commercial PDFlib packages with PHP on platforms with DSO support. PDFlib GmbH makes available packages with precompiled binary PDFlib DSOs for several platforms and PHP versions. If such a package is available for your combination of platform and PHP proceed with Section 5, »Deploying the PDFlib DSO«.

If a PDFlib DSO is not available you need the appropriate precompiled PDFlib C library for your platform (available for all platforms supported by PDFlib GmbH), and must build a PDFlib DSO for PHP according to Section 6, »Building a DSO from a PDFlib C library«.

Using PDFlib Lite with PHP on platforms with DSO support. In order to use PDFlib Lite you need PDFlib source code, and must proceed according to Section 7, »Building PDFlib Lite from Source«. Once you've done this you must build a PDFlib DSO for PHP according to Section 6, »Building a DSO from a PDFlib C library«.

2.2 Platforms where PHP does not reliably support DSOs

In our experience PHP does not support dynamic loading of extensions on the following platforms:

- ▶ IBM AIX
- ▶ Mac OS X: some restrictions apply, see Section 2.1, »Platforms where PHP reliably supports DSOs« above.

Using commercial PDFlib packages with PHP on platforms without DSO support. You need the appropriate precompiled PDFlib C library for your platform, and must rebuild PHP according to Section 8, »Building PHP with statically linked PDFlib«.

Using PDFlib Lite with PHP on platforms without DSO support. In order to use PDFlib Lite you need PDFlib Lite source code and must first build a PDFlib Lite C library according to Section 7, »Building PDFlib Lite from Source«. Next, you must rebuild PHP according to Section 8, »Building PHP with statically linked PDFlib«.

2.3 Other Platforms

There may be other platforms which are supported by PHP, but where we don't know about the status of DSO support. PDFlib GmbH does not support the use of PDFlib with PHP on any of these platforms.

You may be able to make PDFlib with PHP work on such a platform by rolling your own PHP binary. However, be warned that this requires high skills and motivation, and may nevertheless result in frustration instead of working software. After all, there are good reasons why we don't support certain combinations.

2.4 Required Skill Levels

Making PDFlib work with PHP requires various skill levels depending on your operating system platform. We will classify tasks according to the following skill sets:

- ▶ A *PHP Web programmer* knows how to write code for PHP, but doesn't have experience with other languages or general system administration tasks. The PHP programmer usually has access to other people who are responsible for performing configuration tasks.

- ▶ A *sysadmin* feels comfortable working with PEAR and other command-line tools, happily edits *php.ini* and does not hesitate to restart the Web server (i.e. Apache or IIS) if required for installation or configuration purposes. Appropriate permissions (access rights) to do all this are also part of the *sysadmin* profile.
- ▶ A *C developer* has access to a C development environment (header files, compiler, linker, associated system libraries) and can work with configure scripts and Make-files or corresponding IDE features.

It may help to classify yourself according to these types of developers. The remainder of this document describes tasks which require at least sysadmin or C developer skills. PHP developers without additional knowledge or assistance will not be able to perform the required steps without assistance.

3 Testing your Installation

After you installed your PDFlib product extension for PHP using any of the methods discussed in this document you may want to test your installation in order to see whether everything works as expected.

The PHP info page. You can test the success of your PDFlib product installation and configuration with the following mini PHP script:

```
<?phpinfo()?>
```

If you don't find a PDF section check your log files to find the reason. If the output created by *phpinfo()* contains the line *PDFlib GmbH Binary Version* you are using a precompiled PDFlib DSO provided by PDFlib GmbH. If you see a line *PDFlib GmbH Version* you are using your own PDFlib DSO or custom PHP with a statically linked PDFlib. The version number of the PECL module which has been used to build the PDFlib extension will also be shown.

*Note If the output of *phpinfo()* shows only »PDFlib Version« you are using an old PDFlib extension wrapper from a PHP 4 distribution. This combination is not supported.*

The PDFlib product examples. The distribution package of your PDFlib product includes two flavors of examples which you can use to test your installation. In the *bind/php4* directory you will find the old-style examples using the functional API. These examples can be used for PHP 4 and PHP 5. In the *bind/php5* directory you can find the new object-oriented examples. These will work only with PHP 5 or above. To use the examples proceed as follows:

- ▶ Copy some php and data (if available) files to your *htdocs* directory:

```
$ cp bind/php[45]/*.php ../htdocs
$ cp bind/data/* ../htdocs/data
```

- ▶ point your browser to the URLs of the examples
- ▶ enjoy the generated PDFs

4 PDFlib in Hosting Environments

You are running a site at a web hosting provider. In this case there are various considerations (we can ignore the case where a PDFlib extension for PHP is already installed since there's nothing more to do):

- ▶ Some providers do not allow custom PHP extensions; in this case you are out of luck.
- ▶ With some providers you can maintain your own copy of *php.ini*, while others don't allow this. If you can't edit *php.ini* and this file contains *enable_dl=Off* you are out of luck.

You are a web hosting provider. As a provider you should be aware of the following:

- ▶ Although PDFlib Lite source code is freely available, and many Linux and PHP distributions contain PDFlib Lite, the PDFlib Lite license does not cover free use of PDFlib Lite on a Web hoster's systems.
- ▶ You can install commercial PDFlib DSOs even without obtaining a license. In this situation you can install one of the precompiled PDFlib DSOs supplied by PDFlib GmbH without a license key (i.e. a demo stamp will be created). Those among your customers who wish to commercially use it can obtain a commercial license to disable the demo stamp. In other words, you can offer PDFlib without the need for obtaining a license for all of your servers. The recommended method is to install the PDFlib DSO in some globally accessible directory, and set the *extension=* line in *php.ini* appropriately.
- ▶ Alternatively, if (like an increasing number of providers) you believe in PDFlib availability as a competitive advantage, you can obtain a site license which covers all your servers and customers. Individual users will no longer be required to obtain a license on their own in this case. Please contact PDFlib GmbH if you are interested in more details.

5 Deploying the PDFlib DSO

Note In addition to the PDFlib product family, this section also applies to PDFlib TET, PDFlib PLOP, and PDFlib pCOS if you replace the string »libpdf_php« with »libtet_php«, »libplop_php«, or »libpcos_php«, respectively.

Requirements:

- ▶ Skill level: sysadmin
- ▶ The PDFlib DSO, either built on your own or (preferably) from a binary package provided by PDFlib GmbH at www.pdflib.com/download/pdflib-family/pdflib-7
- ▶ Working PHP binary

This section applies to the prebuilt DSOs distributed by PDFlib GmbH, as well as to DSOs which you have built yourself.

5.1 Installing the PDFlib DSO on Windows

The PDFlib DSOs for Windows (actually DLLs) have been tested with the binary PHP distribution which is available from www.php.net. You will find PDFlib DSOs for various versions of PHP on Windows in the following location of the uncompressed package:

```
bind/php[45]/php-<version>/libpdf_php.dll
```

For the PHP installation process please follow the documentation of your PHP distribution, and copy the PDFlib DSO to the directory which is specified in the *extension_dir* line in *php.ini*.

The Windows version of the PDFlib DSO has been built for a multithreaded version of PHP.

5.2 Installing the PDFlib DSO on Unix

The PDFlib DSOs for various Unix platforms are available for different versions of PHP. You will find PDFlib DSOs in the following location of the uncompressed package:

```
bind/php[45]/php-<version>/libpdf_php.so (adjust the shared library suffix if necessary)
```

Copy the PDFlib DSO to the directory which is specified in the *extension_dir* line in *php.ini*.

The Unix versions of the PDFlib DSO have been built without multithread support. However, *libpdf_php_mt.so*, which is available for some Unix platforms, has been built for versions of PHP with experimental Zend Thread Safety (ZTS).

5.3 Using the PDFlib DSO

Loading the PDFlib DSO in *php.ini*. If you decide to load PDFlib every time PHP starts, insert one line in *php.ini*

```
extension=libpdf_php.dll (on Windows)
```

or

`extension=libpdf_php.so` (on Unix; adjust the shared library suffix if necessary)

and restart your Web server so that the changes are recognized.

Loading the PDFlib DSO explicitly in your PHP script. Without the *extension* line in *php.ini* you must include the following line in your PHP scripts:

```
dl("libpdf_php.dll");
```

 (on Windows)

or

```
dl("libpdf_php.so");
```

 (on Unix; adjust the shared library suffix if necessary)

In this case your *php.ini* must contain the following lines:

```
safe_mode=Off  
enable_dl=On
```

The line *extension_dir* is not relevant in this case. Note that for security reasons this method is no longer recommended; many Web hosts do not allow it.

5.4 Common Problems with PDFlib DSOs

Older version of PDFlib built into the PHP binary. PDFlib support must not already have been compiled into your PHP version. If your PHP already includes PDFlib support (this is the case for versions of PHP distributed with some Linux distributions) but you need a newer PDFlib version you must first obtain a PHP binary without builtin PDFlib support (either by locating the appropriate binary, or rebuilding it yourself. This works similar to Section 8, »Building PHP with statically linked PDFlib«, but you must rebuild PHP with the following *configure* option:

```
-with-pdflib=no
```

Note Maintainers of Linux and PHP distributions should include PDFlib support for PHP as DSO because this facilitates updates.

Binary characteristics of PHP and PDFlib DSO must match. Several properties of your PHP binary must match those of the PDFlib DSO. These properties are determined when building PHP, and cannot be changed afterwards. The precompiled DSOs for PDFlib have been built as follows:

- ▶ non-debug version
- ▶ thread-safety as described in Section 5.1, »Installing the PDFlib DSO on Windows« and Section 5.2, »Installing the PDFlib DSO on Unix«
- ▶ the API version: choose the matching version from *bind/php[45]/php-<version>*

If you see an error message similar to the following when trying to load the PDFlib DSO, your PHP build number does not match that of the PDFlib module:

```
Warning: pdf: Unable to initialize module  
Module compiled with debug=0, thread-safety=0 module API=20020429  
PHP compiled with debug=0, thread-safety=1 module API=20020429
```

All of these options must match.

PDFlib with XAMPP or MAMP on Mac OS X. If you add the PDFlib PHP extension to your *php.ini* on a Mac OS X Intel box with XAMPP Mac OS X 0.6.3 installed, the following error message appears:

```
dyld: NSLinkModule() error
dyld: Symbol not found: __cg_jpeg_resync_to_restart
Referenced from: /System/Library/Frameworks/ApplicationServices.framework/Versions/A/
Frameworks/ImageIO.framework/Versions/A/ImageIO
Expected in: /Applications/xampp/xamppfiles/lib/libjpeg.62.dylib
```

The PDFlib extension is linked against the *ApplicationServices* Framework, and XAMPP changes the `DYLD_LIBRARY_PATH`. This combination confuses the dynamic link editor. We detected that commenting out `DYLD_LIBRARY_PATH` in *xamppfiles/bin/envvars* cures this problem.

A similar problem arises with MAMP. To cure the problem with MAMP comment out `DYLD_LIBRARY_PATH` in *Library/bin/envvars*.

6 Building a DSO from a PDFlib C library

Requirements:

- ▶ Skill level: sysadmin
- ▶ a precompiled PDFlib C library, either built on your own from PDFlib Lite source code or (preferably) from a binary package provided by PDFlib GmbH at www.pdflib.com/download/pdflib-family/pdflib-7
- ▶ PECL package for PDFlib, available from pecl.php.net/package/pdflib
- ▶ PEAR, available from pear.php.net

6.1 Building the PDFlib DSO with PEAR

The simplest way to build a PDFlib extension for PHP is to use the PECL module for PDFlib, and build it with the *pear* command provided by PHP. PEAR (PHP Extension and Application Repository) and PECL (a repository for PHP Extensions) are part of the PHP project. The *pear* tool is included with PHP 4.3.0 and above, but can also be installed for older versions. If you don't have *pear* installed with your PHP installation, or it does not work properly for some reason you should try to get the latest version of *pear*.

To build and install the PDFlib extension use the following call on Unix:

```
pear install pdflib
```

or the following call on Windows:

```
pear install pecl.php.net/pdflib
```

and supply the path where PDFlib is installed at the following prompt:

```
path to pdflib installation? : $HOME/pdflib-7.0.x/bind/c
```

This will create a PDFlib module called *pdf.so* (or similar, depending on platform). After building the extension you should check your *php.ini* to make sure that the extension was added properly, and restart Apache to activate the new module.

Other issues with PECL/PEAR and PDFlib.

- ▶ Trying to build the PECL package with GD support may fail, since PHP does not yet support building PECL packages which use GD. Until this PHP issue is resolved you have to copy the GD header files manually to */usr/include/php* (or wherever the PHP header files are located in your installation). See https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=145891 for more details
- ▶ Using PECL with PHP 4 on Mac OS X leads to incomplete DSOs. The build process in this situation silently ignores the *PHP_ADD_FRAMEWORK(ApplicationServices)* directive in the configuration files. As a result, the DSO is not linked against this framework and PDFlib's hostfont feature (access to system-installed fonts) will not work.

6.2 Building the PDFlib DSO without PEAR

Although it is possible to build a PDFlib DSO without PEAR, this method is not supported, and not described here.

7 Building PDFlib Lite from Source

Requirements:

- ▶ Skill level: C developer
- ▶ PDFlib Lite source code, available from www.pdflib.com/download/free-software/pdflib-lite

On Unix systems (for Windows see *readme-source-windows.txt* in the PDFlib Lite distribution) unpack the PDFlib Lite source code, change to the created directory, and issue the following commands:

```
$ ./configure
$ make
$ make install
```

This will create a PDFlib Lite C library. For more details see *readme-source-unix.txt* in the PDFlib Lite distribution.

The last step in the sequence above will install the library and include files to some local directory. By default these files will be installed to some system directory, which means that sysadmin permissions are required. However, you can also install to some private (non-privileged) directory by using the *--prefix* option in the initial call to the *configure* script, e.g.

```
$ ./configure --prefix=$HOME
```

Note that the installation will automatically create *libs* and *include* subdirectories under the specified prefix.

8 Building PHP with statically linked PDFlib

Requirements:

- ▶ Skill level: C developer, preferably with experience in building large software systems, plus sysadmin
- ▶ a precompiled PDFlib C library, either built on your own from PDFlib Lite source code or (preferably) from a binary package provided by PDFlib GmbH at www.pdflib.com/download/pdflib-family/pdflib-7
- ▶ PHP source code, available from www.php.net/downloads.php
- ▶ PECL package for PDFlib, available from pecl.php.net/package/pdflib

If DSOs don't work in your situation you must rebuild PHP from source code in order to include PDFlib support. This step must be done within the PHP build process. Below you will find an overview on how to achieve this, but you'll have to refer the PHP documentation for more details.

Building PHP. Unpack PHP, configure and build it to your needs. We recommend to initially try this without PDFlib support in order to get accustomed with the process.

Prepare the PDFlib C library. For the commercial PDFlib product: unpack the PDFlib C library and note the full path to the *bind/c* directory in this package.

For PDFlib Lite: build the PDFlib Lite C library according to Section 7, »Building PDFlib Lite from Source«.

Prepare the PECL Package. Create an *ext/pdf* directory in your PHP source tree. Unpack the PECL package for PDFlib, and move the contents of this package (including *pdf.c*, *php_pdf.h* and others) to the *ext/pdf* directory of your PHP source tree.

Put everything together. Change to the PHP source directory and follow these steps:

```
# first remove the configure script so that buildconf is forced to create it
$ rm ./configure

# rebuild the PHP configure script:
$ ./buildconf --force

# configure and build PHP with PDFlib support:
$ ./configure --with-pdflib=<pdflib-source-directory>/bind/c <other-php-options>
....
Thank you for using PHP.
config.status: creating php5.spec
config.status: creating main/build-defs.h
config.status: creating scripts/phpize
config.status: creating scripts/php-config
config.status: creating sapi/cli/php.1
config.status: creating main/php_config.h
config.status: executing default commands

$ make
....
Build complete.
(It is safe to ignore warnings about tempnam and tmpnam).

# install it as root
```

```
$ sudo make install
```

```
# probably you have to configure your Web server to load the new php module
```

```
# restart apache (i.e. sudo apachectl restart)
```

Notes for Mac OS X. As mentioned in Section 2.1, »Platforms where PHP reliably supports DSOs« only some combinations of PHP versions and versions of Mac OS X reliably support DSOs. Other combination may require some manual tweaking where the following notes may be useful:

- ▶ On Mac OS X *pear install* will fail if no *libtoolize* is available. Either install it from the Apple Developer Tools CD, or try to change *libtoolize* to *glibtoolize* in the *phpize* script on your system.
- ▶ If the *configure* script fails on Mac OS X you must set the *LDFLAGS* environment variable appropriately. Use the following commands in the C shell (*tcsh* or *csh*):

```
$ setenv LDFLAGS "-framework ApplicationServices"
```

In the Bourne shell use the following commands:

```
$ LDFLAGS='-framework ApplicationServices'
```

```
$ export LDFLAGS
```

- ▶ More information regarding PHP on Mac OS X can be found here:
www.entropy.ch/software/macosx/php
tech.groups.yahoo.com/group/pdflib/message/11347

9 Additional Web Links

- ▶ The public PDFlib mailing list for general discussion:
tech.groups.yahoo.com/group/pdflib
- ▶ PDFlib support for commercial licensees:
support@pdflib.com
- ▶ General information on installing PHP:
www.php.net/install
- ▶ PEAR and PECL support:
pear.php.net/support.php and pecl.php.net/support.php
- ▶ Instructions on getting the latest version of PEAR:
pear.php.net/manual/en/installation.getting.php
- ▶ Comprehensive list of PHP-related links:
www.php.net/links.php