

How to use PDFlib products with PHP

Last change: April 14, 2010

Latest PDFlib version covered in this document: 8.0.1

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:

- ▶ 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 the products PDFlib TET (Text Extraction Toolkit), PDFlib PLOP (Linearization, Optimization, Protection), and PDFlib pCOS (PDF Information Retrieval Tool) are included where applicable.

Note This document does not apply to PDFlib Lite, the open-source subset of PDFlib 7.

The methods for deploying any of these products as a PHP extension are the same in all cases. 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 8.0.1 and PDFlib 7.0.5
- ▶ TET 3.0, PLOP 4.0, and pCOS 2.0
- ▶ PHP 5.1.x, 5.2.x, and 5.3.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 10.6 (see Section 2.2, »Platforms where PHP does not reliably support DSOs« for older versions)
- ▶ Linux on x86 and ia64
- ▶ Linux on zSeries
- ▶ FreeBSD 6/7 on x86
- ▶ Sun Solaris 10 on x86 and Sparc
- ▶ AIX 5/6 32-bit
- ▶ HP-UX 11

The PDFlib 8.0.1 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/php/php-510* for PHP 5.1.0 – 5.1.6
- ▶ *bind/php/php-520* for PHP 5.2.0 – 5.2.13
- ▶ *bind/php/php-530* for PHP 5.3.0 – 5.3.2

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.

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«.

2.2 Platforms where PHP does not reliably support DSOs

PHP on Mac OS X 10.5 and earlier. The PDFlib DSOs works fine with the native PHP version bundled with Mac OS X 10.6 (Snow Leopard). However, Apple's PHP version which is bundled with Mac OS X 10.4 and 10.5 does not work with PDFlib DSOs. To use PHP with PDFlib on Mac OS X 10.5 and earlier you need third-party PHP packages such as MAMP or XAMPP for Mac.

Mac OS X 10.5 (Leopard) suffers from an additional restriction: 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 may 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");
```

2.3 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 Makefiles 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.

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/php* directory you can find PDFlib programming examples. To use the examples proceed as follows:

- ▶ Copy the PHP samples and data files to your *htdocs* directory:

```
$ cp bind/php/*.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-8
- ▶ 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 uncompressed package.

The PDFlib DSO in the following directory has been built for a multithreaded version of PHP:

```
bind/php/php-<version>/libpdf_php.dll
```

Starting with PHP 5.2 we also offer Windows binaries of the PDFlib PHP binding which have been built without support for thread safety. These binaries are named as follows (ZTS refers to *Zend Thread Safety*, a threading abstraction layer):

```
bind/php/php-<version>-nozts/libpdf_php.dll
```

Since PHP 5.3 the precompiled PHP binaries for Windows (*php.exe*) come in two incompatible flavors: one is compiled with Visual Studio 6, while the other is compiled with Visual Studio 2008 (also called VS9). The VS9 versions of the PDFlib PHP binding are located in directories ending with »_VS9«, e.g.

```
bind/php/php-<version>_VS9/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*.

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/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 standard Unix versions of the PDFlib DSO have been built without multithread support. However, the binaries

```
bind/php/php-<version>mt/libpdf_php.so
```

which are available for some platforms, are PDFlib DSOs for use with versions of PHP which include *Zend Thread Safety* (ZTS) support.

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 hosters do not allow it.

5.4 Common Problems with PDFlib DSOs

Older version of PDFlib built into the PHP binary. PDFlib Lite support must not already have been compiled into your PHP version. If your PHP already includes PDFlib Lite 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).

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/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 10.5 and older. If you add the PDFlib PHP extension to your *php.ini* on a Mac OS X Intel machine which has XAMPP 1.7.2a 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 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