# How to use PDFlib products with PHP

Last change: December 11, 2017
Latest PDFlib version covered in this document: 9.1.2
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 products as a PHP extension. The generic term PDFlib is used to designate one of the following distinct products:

► The PDFlib base 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.

Most of the PDFlib information applies to other PDFlib GmbH products analogously. Notes for the following products are included where applicable:

► PDFlib TET (Text and Image Extraction Toolkit)
► PDFlib PLOP (Linearization, Optimization, Protection) and PLOP DS (Digital Signature)

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. Multiple products can coexist within one PHP installation, however. Note that the evaluation versions of commercial PDFlib products are fully functional, but display a demo stamp across all generated PDF pages unless a valid license key is applied. Other PDFlib GmbH products have other restrictions in evaluation mode (see documentation).

This document applies to the following versions of PDFlib GmbH products:

► PDFlib 9.1.2
► TET 5.1
► PLOP and PLOP DS 5.2

Where applicable, version-specific information is provided separately.

# 2 Platforms, PHP Versions and Thread Safety

**Supported platforms.** 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):

- ► Windows Server x86/x64 and Windows XP/Vista/7/8/10 x86/x64;
- ► OS X/macOS
- ► Linux x86 and Intel 64
- ► Linux on zSeries
- ► FreeBSD x86/Intel 64
- ► Oracle Solaris on x86 and Sparc
- ► HP-UX IA-64
- ► AIX 32-bit; the PDFlib DSO for AIX can also be used on IBM i5/iSeries with PASE
- ► IBM i5/iSeries (see Section 6.4, »Installing and Using the PDFlib DSO on i5/iSeries«)

**Supported PHP versions.** The distribution packages shipped by PDFlib GmbH contain 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-540* for PHP 5.4.0 and above
- ► *bind/php/php-550* for PHP 5.5.0 and above
- ► *bind/php/php-560* for PHP 5.6.0 and above
- ► *bind/php/php-700* for PHP 7.0.0 and above
- ► *bind/php/php-710* for PHP 7.1.0 and above
- ► *bind/php/php-720* for PHP 7.2.0 and above

**Thread safety.** On most platforms two variants of the PHP binary are available, and the PDFlib DSO must match the selected PHP version:

- ► Thread-safe (TS): this version should generally be used when loading PHP as a module into a Web server. The thread-safe version should be used when loading PHP as an Apache module.
- ► Non-thread-safe (NTS): this version includes *nts* in the file name and should generally be used when using PHP in a Web server through FastCGI protocol or when working with the command-line interface (CLI). The NTS version should be used when integration PHP with IIS via FastCGI.

Refer to the PHP Web site for more information regarding TS and NTS versions of PHP.

# 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.

# 4 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()?>
```

Check the output created by *phpinfo( )* for one of the following:
- ► If the output contains the line

  ```
  PDFlib GmbH Binary Version
  ```

  you are using a precompiled PDFlib DSO provided by PDFlib GmbH.
- ► If you see the 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.
- ► If you don't find any PDFlib section check your log files to determine the reason.

**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 or another directory which has been configured appropriately in the Web server:

  ```
  $ cp bind/php /path/to/htdocs
  $ cp bind/data /path/to/htdocs
  ```

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

# 5 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.

# 6 Deploying the PDFlib DSO

*Note* *In addition to the PDFlib product family, this section also applies to PDFlib TET and PDFlib PLOP if you replace the string* `php_pdflib` *with* `php_tet` *or* `php_plop`*.*

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-9*
- ► Working PHP binary

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

## 6.1 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 *windows.php.net*. You will find PDFlib DSOs for various versions of PHP on Windows with and without multi-threading support in the distribution package.

Depending on the target PHP version the PDFlib DSOs have been built with different versions of Visual Studio, which means that the corresponding redistributable runtime DLLs must be available on the system:
- ► The PDFlib DSOs for PHP 5.4 have been built with Visual Studio 2008 (VC 9).
- ► The thread-safe PDFlib DSOs for PHP 5.5 and 5.6 have been built with Visual Studio 2012 (VC 11). The non-thread-safe DSO for PHP 5.5 is also available in a variant built with Visual Studio 2008 (VC 9).
- ► The PDFlib DSOs for PHP 7.0 and 7.1 have been built with Visual Studio 2015 (VC 14).
- ► The PDFlib DSOs for PHP 7.2 have been built with Visual Studio 2017 (VC 15).

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*.

**Using PDFlib with Zend Server.** In order to use PDFlib with Zend Server you must install the DLL *php_pdflib.dll* from the *php-<version>-nozts_VS9* directory. Copy this DLL to the extension directory and restart PHP.

The PDFlib module for Zend Server has been built with Visual Studio 2008 (VS9).

## 6.2 Installing the PDFlib DSO on Unix

The PDFlib DSOs for various Unix platforms with and without multi-threading support are available for different versions of PHP. You will find PDFlib DSOs in the following location of the distribution package (adjust the shared library suffix as necessary for your platform):

```
bind/php/php-<version>/php_pdflib.so
```

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

**Using PDFlib with Zend Server.**    In order to use PDFlib with Zend Server you must install the DSO *php_pdflib.so* from the *php-<version>* directory. Copy this DSO to the extension directory and restart PHP.

## 6.3 Using the PDFlib DSO

**Loading the PDFlib DSO in php.ini.**    If you decide to load the PDFlib DSO every time PHP starts, insert the following line in *php.ini* (adjust the shared library suffix *.dll* as necessary for your platform, e.g. *.so):*

```
extension=php_pdflib.dll
```

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 (adjust the shared library suffix *.dll* as necessary for your platform, e.g. *.so):*

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

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 and many Web hosters don't allow it.

**Custom pdflib.ini file.**    If your PHP version supports additional *.ini* files parsing you can create a *pdflib.ini* configuration file with a single line as follows (adjust the shared library suffix *.dll* as necessary for your platform, e.g. *.so):*

```
extension=php_pdflib.dll
```

in the *PHP_INI_SCAN_DIR* directory.

**Using PDFlib with Zend Server.**    Zend automatically recognizes correctly named extensions when PHP is restarted.

## 6.4 Installing and Using the PDFlib DSO on i5/iSeries

In order to use PDFlib, TET or PLOP/PLOP DS with PHP on IBM i5/iSeries you need AIX binaries of PDFlib GmbH products. The required product binaries are not included in the i5 packages, but must be taken from the corresponding AIX package. Although you must download AIX product packages in the situations described above, you must specify IBM i5 as target platform when ordering license keys for using AIX binaries on an i5 system. PDFlib GmbH product binaries for AIX accept i5 license keys when running with PASE on i5/OS.

The third-party product Zend Server for IBM i allows you to »leverage the power of the IBM i platform and the strength and flexibility of PHP to run business-critical applications on IBM i«, see

```
www.zend.com/en/solutions/modernize-ibm-i
```

The requirements for using PDFlib with PHP on i5/iSeries are as follows:
► Zend Server for IBM i or Zend Server Community Edition (CE) for IBM i
► PHP 5.4 or above

Zend Server for IBM i is based on the Portable Application Solutions Environment (PASE for i), an »integrated runtime for porting selected applications from AIX systems«. PASE is not an emulation: since i5/iSeries and AIX are based on the same POWER processor architecture, PASE uses the processor directly. There are no performance disadvantages when using PASE. More details about PASE can be found on the following IBM Web site:

```
www.ibm.com/partnerworld/wps/servlet/ContentHandler/pw_com_porting_ibmi_pase
```

Perform the following steps to use PDFlib with Zend Server for IBM i:
► Since Zend Server and the underlying Apache Web server are based on the PASE environment, you must use the PDFlib package for AIX, not the PDFlib package for i5/iSeries. Download the following package from the PDFlib Web site:

```
PDFlib-9.x.y-AIX-php.tar.gz
```

► Unpack the PDFlib package for AIX, using one of the available tools for unpacking *.tar.gz* packages.
► Locate *php_pdflib.so* and copy it to the *extension_dir* of Zend Server. The output of *phpinfo( )* shows the exact location of this directory.
► As an alternative to the previous step you can also load the PDFlib DSO directly from your script without configuring it in *php.ini* (note that for security reasons this method is no longer recommended):

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

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

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

Now you can create PDF from PHP scripts on i5/iSeries.

## 6.5 Common Problems with PDFlib DSOs

### 6.5.1 All Platforms

**Binary characteristics of PHP and PDFlib DSO must match.**    Several properties of your PHP binary must match the corresponding properties 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
► with or without multi-threading support (see »Thread safety«, page 2)
► 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:

```
PHP Warning: PHP Startup: PDFlib: Unable to initialize module
Module compiled with build ID=API20100525,NTS
PHP compiled with build ID=API20100525,TS
```

All of these options must match. In the example above it was attempted to load the non-thread-safe DSO into a thread-safe PHP binary.

## 6.5.2 Linux x86 and Intel 64

**PDFlib with XAMPP on Linux x86.**    Some versions of system libraries bundled with the XAMPP package may trigger the following error message:

```
Warning: PHP Startup: Unable to load dynamic library '/opt/lampp/htdocs/test/pdf/pdflib/
bind/php/php-530/php_pdflib.so' - /opt/lampp/lib/libgcc_s.so.1: version `GCC_4.2.0' not
found (required by /usr/lib/libstdc++.so.6) in Unknown on line 0
```

In this case you must disable the following two lines in the file *bin/envvars*, e.g. by adding a comment character at the start of the line:

```
#binbuild LD_LIBRARY_PATH="/opt/lampp/lib/:$LD_LIBRARY_PATH"
#binbuild export LD_LIBRARY_PATH
```

**PDFlib with XAMPP on Linux Intel 64.**    If you are using the 32-bit edition of XAMPP you must use the 32-bit edition of PDFlib for this combination. However, you may see the following error message:

```
Warning: PHP Startup: Unable to load dynamic library
'/opt/lampp/htdocs/test/pdf/PDFlib-x.y.z-Linux-php/bind/php/php-710/php_pdflib.so'
- libstdc++.so.6: wrong ELF class: ELFCLASS64 in Unknown on line 0
```

The reason for this error is that while XAMPP includes some of the 32-bit runtime libraries required for PDFlib, one important runtime library is still missing. You must install the 32-bit version of *libstdc++.so.6* on the system. For example, on Debian systems this can be achieved with the following command:

```
apt-get install ia32-libs
```

## 6.5.3 OS X/macOS

**Problems with host font access and certain combinations of PHP and OS X/macOS.**
Generally the PDFlib DSO works fine with the native PHP version bundled with OS X. However, there are subtle incompatibilities which affect certain combinations of PHP and OS X. So far we are aware of the following affected combinations:
- MAMP and OS X 10.9
- Zend Server and OS X (any version)

In these environments the server may hang or crash when attempting to use PDFlib to create text output. If the server hangs you will find the following message in the server log:

```
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.
```

An Apache crash may be reported in the dump as follows:

```
Process: httpd [14885]
Path: /Applications/MAMP/*/httpd
...
Exception Type: EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: KERN_INVALID_ADDRESS at 0x0000000000000008
```

In order to avoid this problem you must disable the host font feature in PDFlib using the following call:

```
p.set_option("usehostfonts=false");
```

**PDFlib with XAMPP on OS X/macOS.**    If you add the PDFlib PHP extension to your *php.ini* on an OS X/macOS Intel machine which has XAMPP 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 found that unsetting *DYLD_LIBRARY_PATH* cures this problem. Use the following line as last command in *xamppfiles/bin/envvars:*

```
unset DYLD_LIBRARY_PATH
```

# 7 Additional Web Links

▶ The public PDFlib mailing list for general discussion:
*groups.yahoo.com/neo/groups/pdflib/conversations/topics*
▶ 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*