

How to use PDFlib products with PHP

Last change: June 21, 2021

Latest PDFlib version covered in this document: 9.3.1

Latest version of this document available at:

www.pdflib.com/documentation/howtos/

Contact:

PDFlib GmbH

Franziska-Bilek-Weg 9

80339 München, Germany

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 superset of PDFlib which 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. However, multiple products can coexist within one PHP installation. Evaluation versions of 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 9.3.1
- ▶ TET 5.3
- ▶ PLOP and PLOP DS 5.4

Where applicable, version-specific information is provided separately.

2 Platforms, PHP Versions and Thread Safety

Supported platforms. PDFlib GmbH products for use with PHP are as loadable extension modules which are implemented as DSOs (dynamic shared objects, also called dynamic link library DLL). They are available on the following platforms:

- ▶ Windows Server on x86/x64 and Windows 8/10 on x86/x64
- ▶ macOS x64/ARM64
- ▶ Linux x86/x64/ARM64
- ▶ Alpine Linux on x64
- ▶ Linux on zSeries
- ▶ FreeBSD x64

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

- ▶ *bind/php/php-730* for PHP 7.3.x
- ▶ *bind/php/php-740* for PHP 7.4.x
- ▶ *bind/php/php-800* for PHP 8.0.x

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 directory name and should generally be used when using PHP in a Web server through FastCGI protocol, working with the command-line interface (CLI), or integrating 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 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 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, 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 small PHP script:

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

Check the output created by *phpinfo()*. If it contains the line

PHP extension built by PDFlib GmbH

you are using a PDFlib DSO provided by PDFlib GmbH. 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 contains examples in the *bind/php* directory. You can use them to test your installation 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 -r bind/php /path/to/htdocs
cp -r 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 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 from a package provided by PDFlib GmbH at www.pdflib.com/download/pdflib-product-family/
- Working PHP installation

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:

- The PDFlib DSOs for PHP 7.3 and 7.4 have been built with Visual Studio 2017 (VC 15).
- The PDFlib DSOs for PHP 8.0 have been built with Visual Studio 2019 (VC 16).

The corresponding redistributable runtime DLLs must be installed on the system. These Microsoft packages can be found at the following locations:

- Runtime for Visual Studio 2015-2019 on x86: aka.ms/vs/16/release/VC_redist.x86.exe
- Runtime for Visual Studio 2015-2019 on x64: aka.ms/vs/16/release/VC_redist.x64.exe

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`.

6.2 Installing the PDFlib DSO on Linux/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 Installing the PDFlib DSO on macOS with SIP

The general installation procedure for macOS is the same as on Unix systems (see above). However, there are some additional aspects when Apple's *System Integrity*

Protection (SIP) is involved. When SIP is enabled the PHP binary which comes preinstalled with macOS 10.14 Mojave accepts only signed extensions and extensions must be loaded from one of the protected system directories for libraries. Other extension directories are not accepted. The PDFlib DSO is signed by PDFlib GmbH, but installation requires some extra steps on systems with active SIP. These steps are not required for other builds of the PHP binary, e.g. those from Homebrew.

Step 1: Temporarily disable System Integrity Protection (SIP). This requires rebooting into recovery mode:

- ▶ Restart the system and press cmd-R until the Apple logo appears.
- ▶ Select *Terminal* from the *Utilities* menu.
- ▶ In the window that opens type *csrutil disable* and press return to disable System Integrity Protection.
- ▶ Choose *Restart* from the Apple menu.

Step 2: Install the PDFlib DSO in the PHP extension directory. Copy the appropriate version of the PDFlib DSO to the extension directory of the preinstalled PHP binary, e.g.

```
cp bind/php/php-710-nts/php_pdflib.so /usr/lib/php/extensions/no-debug-non-zts-20160303/
```

Step 3: Enable System Integrity Protection again. This requires rebooting again:

- ▶ Restart the system and press cmd-R until the Apple logo appears.
- ▶ Select *Terminal* from the *Utilities* menu.
- ▶ In the window that opens type *csrutil enable* and press return to enable System Integrity Protection. This ensures that your machine is fully protected by SIP.
- ▶ Choose *Restart* from the Apple menu.

Now configure the PDFlib extension for PHP with one of the methods described in the next section.

6.4 Using the PDFlib DSO

Loading the PDFlib DSO in *php.ini*. Insert the following line in *php.ini*:

```
extension=php_pdflib
```

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

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:

```
extension=php_pdflib
```

in the *PHP_INI_SCAN_DIR* directory.

6.5 Common Problems with PDFlib DSOs

6.5.1 All Platforms

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 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:

```
Warning: PHP Startup: PDFlib: Unable to initialize module
Module compiled with build ID=API20190902,TS
PHP    compiled with build ID=API20190902,NTS
These options need to match
in Unknown on line 0
```

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 on x86 and x64

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-720/php_pdfplib.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 x64. 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-720/php_pdfplib.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 macOS

PDFlib with XAMPP on macOS. If you add the PDFlib PHP extension to your *php.ini* on a macOS x64 system 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
```