How to use PDFlib products with PHP

Last change: June 17, 2013
Latest PDFlib version covered in this document: 9.0.1
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 products 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.

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

► PDFlib TET (Text Extraction Toolkit)
► PDFlib PLOP (Linearization, Optimization, Protection) and PLOP DS (Digital Signature)
► PDFlib pCOS (PDF Information Retrieval Tool)

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. Other PDFlib GmbH products have other restrictions in evaluation mode (see documentation).

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

► PDFlib 8.0.6 and PDFlib 9.0.1 (use appropriate version number 8 or 9 in path names)
► TET 4.2
► PLOP and PLOP DS 4.1
► pCOS 3.0

Where applicable, version-specific information is provided separately.

2 Supported Platforms and PHP Versions

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 plat-forms (only platforms supported by PDFlib GmbH are mentioned here):

► Windows Server on x86 and Windows XP/Vista/7/8 on x86
► OS X and OS X Server 10.6/10.7/10.8
► Linux on x86 and Intel 64
► Linux on zSeries
► FreeBSD 7/8/9 on x86
► Oracle Solaris 10 on x86 and Sparc
► HP-UX
► AIX 32-bit (the AIX binary can also be used on IBM i5/iSeries)
► IBM i5/iSeries (see Section 6.4, »Installing and Using the PDFlib DSO on i5/iSeries«)

The PDFlib 9 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-530* for PHP 5.3.0 – 5.3.26
► *bind/php/php-540* for PHP 5.4.0 – 5.4.16
► *bind/php/php-550* for PHP 5.5.0 (not available in PDFlib 8)

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 plat-forms and PHP versions. If such a package is available for your combination of platform and PHP proceed with Section 6, »Deploying the PDFlib DSO«.

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:

    ```
    $ cp bind/php/*.php .../htdocs
    $ cp bind/data/* .../htdocs/data
    ```

- 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, PDFlib PLOP, and PDFlib pCOS if you replace the string »php_pdflib« with »php_tet«, »php_plop«, or »libpcos_ php«, respectively.

Requirements:
- ► Skill level: sysadmin
- ► The PDFlib DSO, either built on your own or (preferably) from a binary package pro-vided by PDFlib GmbH at *www.pdflib.com/download/pdflib-family/pdflib-9*
- ► Working PHP binary

This section applies to the prebuilt 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 dis-tribution 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          (PDFlib 8)
bind/php/php-<version>/php_pdflib.dll           (PDFlib 9)
```

We also offer Windows binaries of the PDFlib PHP binding which have been built with-out 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     (PDFlib 8)
bind/php/php-<version>-nozts/php_pdflib.dll      (PDFlib 9)
```

The PDFlib binding for PHP has been built for the Visual Studio 2008 (also called VS9) version.

For the PHP installation process please follow the documentation of your PHP distri-bution 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 9 with Zend Server you must in-stall the DLL *php_pdflib.dll* from the *php-<version>-nozts_VS9* directory. Copy this DLL to the extension directory and restart PHP.

For PDFlib 8 you must rename *libpdf_php.dll* to *php_pdf.dll* to make sure that the ex-tension will automatically be recognized by PHP.

6.2 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 (adjust the shared library suffix as necessary for your platform):

```
bind/php/php-<version>/libpdf_php.so          (PDFlib 8)
bind/php/php-<version>/php_pdflib.so           (PDFlib 9)
```

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         (PDFlib 8)
bind/php/php-<version>mt/php_pdflib.so          (PDFlib 9)
```

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

Using PDFlib with Zend Server.   In order to use PDFlib 9 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.

For PDFlib 8 you must rename *libpdf_php.so* to *php_pdf.so* to make sure that the extension will automatically be recognized by PHP.

6.3 Using the PDFlib DSO

Loading the PDFlib DSO in php.ini.   If you decide to load PDFlib 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=libpdf_php.dll                       (PDFlib 8)
extension=php_pdflib.dll                        (PDFlib 9)
```

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("libpdf_php.dll");                          (PDFlib 8)
dl("php_pdflib.dll");                           (PDFlib 9)
```

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.

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

Note *In order to deploy PDFlib with Zend Server for IBM i you must order a PDFlib license key for AIX, not a license for i5/iSeries. See below for more information.*

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/products/server/zend-server-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 5.3

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-8.0.x-AIX-php.tar.gz        (PDFlib 8)
PDFlib-9.0.x-AIX-php.tar.gz        (PDFlib 9)
```

- ► Unpack the PDFlib package for AIX, using any of the available tools for unpacking *.tar.gz* packages.
- ► Locate *libpdf_php.so* (PDFlib 8) or *php_pdflib.so* (PDFlib 9) and copy it to the *extension_dir* of Zend Server. The output of *phpinfo( )* shows the exact location of this directory.
- ► PDFlib 8 only: rename the copy of *libpdf_php.so* to *php_pdf.so* to match the naming conventions used in Zend Server.
- ► 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 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 6.1, »Installing the PDFlib DSO on Windows« and Section 6.2, »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.

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.

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/libpdf_php.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. Since XAMPP is only available as a 32-bit edition 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-8.0.4-Linux-php/bind/php/php-530/libpdf_php.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

PDFlib with XAMPP or MAMP on OS X 10.6 and older.   If you add the PDFlib PHP extension to your *php.ini* on an OS X 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 commenting out *DYLD_LIBRARY_PATH* in *xamppfiles/bin/envvars* cures this problem.

If you have MAMP installed, the following error message may appear in the log file:

```
PHP Warning: PHP Startup: Unable to load dynamic library '/Applications/MAMP/bin/php5.3/
lib/php/extensions/no-debug-non-zts-20090626/libpdf_php.so' - dlopen(/Applications/MAMP/
bin/php5.3/lib/php/extensions/no-debug-non-zts-20090626/libpdf_php.so, 9): Symbol not
found: __cg_jpeg_resync_to_restart
Referenced from: /System/Library/Frameworks/ApplicationServices.framework/Versions/A/
Frameworks/ImageIO.framework/Versions/A/Resources/libTIFF.dylib
Expected in: /Applications/MAMP/Library/lib/libjpeg.8.dylib
in /System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/
ImageIO.framework/Versions/A/Resources/libTIFF.dylib in Unknown on line 0
```

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

7 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