PHP

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

Contents

[hide]

Installation

Install the **`php`** package from the **official repositories**.

You can find older versions of PHP in the AUR including **`php53`**AUR, **`php55`**AUR, **`php56`**AUR, **`php70`**AUR, and **`php71`**AUR.

Note that to run PHP scripts as plain CGI, you need the **`php-cgi`** package.

Running

While PHP can be run standalone, it is typically used with http servers such as **Apache HTTP Server** (this is frequently called **LAMP** setup), **nginx**, **lighttpd** and **Hiawatha**.

To run PHP standalone issue the `php -S localhost:8000 -t public_html/` command. See **documentation**.

Configuration

The main PHP configuration file is well-documented and located at `/etc/php/php.ini`.

- It is recommended to set your timezone (**list of timezones**) in `/etc/php/php.ini` like so:

```
date.timezone = Europe/Berlin
```

- If you want to display errors to debug your PHP code, change `display_errors` to `On` in `/etc/php/php.ini`:

```
display_errors = On
```

- The **open_basedir** directive limits the paths that can be accessed by PHP, thus increasing security at the expense of potentially interfering with normal program execution. Starting with PHP 7.0, it is **no longer set by default** to more closely match upstream so users who wish to use it must configure it manually. Example:

```
open_basedir = /srv/http/:/home/:/tmp/:/usr/share/pear/:/usr/share/webapps/
```

Extensions

A number of commonly used PHP extensions can also be found in the official repositories:

```
$ pacman -Ss php-
```

Tip: Instead of editing `/etc/php/php.ini`, an extension may be enabled/configured in the `/etc/php/conf.d` directory instead (e.g. `/etc/php/conf.d/gd.ini`)

Extensions for current and older versions of PHP are also available in **AUR** under the prefix php-* and php56-*, e.g. **php-imagick**AUR, **php-redis**AUR, **php56-mcrypt**AUR.

gd

For **php-gd** uncomment the line in `/etc/php/php.ini`:

```
extension=gd
```

Imagemagick

Install the **imagemagick** package and install one of the listed PHP extension library.

Install **php-imagick**AUR, it will create the file `/etc/php/conf.d/imagick.ini` to configure the extension.

PECL

Make sure the **php-pear**AUR package has been installed:

```
# pecl install imagick
```

Create a `/etc/php/conf.d/imagick.ini` and enable the extension:

```
/etc/php/conf.d/imagick.ini
---------------------------
extension=imagick
```

pthreads

If you wish to have POSIX multi-threading you will need the pthreads extension. To install the pthreads (**http://pecl.php.net/package/pthreads**) extension using `pecl` you are required to use a compiled version of PHP with the the thread safety support flag `--enable-maintainer-zts`. Currently the most clean way to do this would be to rebuild the original package with the flag.

Instruction can be found on the **PHP pthreads extension** page.

PCNTL

PCNTL allows you to create process directly into the server side machine. While this may seen as something you would want, it also gives PHP the power to mess things up really badly. So it is a PHP extension that cannot be loaded like other more convenient extension. This is because of the great power it gives to PHP. To enable it PCNTL has to be compiled into PHP.

The php package on Arch Linux is currently built with "--enable-pcntl", so that it should be available by default.

MySQL/MariaDB

Install and configure MySQL/MariaDB as described in **MariaDB**.

Uncomment **the following lines** in `/etc/php/php.ini`:

```
extension=pdo_mysql
extension=mysqli
```

Note: `extension=mysql` was **removed** in PHP 7.0.

You can add minor privileged MySQL users for your web scripts. You might also want to edit `/etc/mysql/my.cnf` and uncomment the `skip-networking` line so the MySQL server is only accessible by the localhost. You have to restart MySQL for changes to take effect.

Redis

Install and configure **Redis**, then install **phpredis-git**AUR.

Uncomment the line of the package, e.g. `/etc/php/conf.d/redis.ini`.

PostgreSQL

Install and configure **PostgreSQL**, then install the **php-pgsql** package and uncomment the following lines in `/etc/php/php.ini`:

```
extension=pdo_pgsql
extension=pgsql
```

Sqlite

Install and configure **SQLite**, then install the **`php-sqlite`** package and uncomment the following lines in `/etc/php/php.ini`:

```
extension=pdo_sqlite
extension=sqlite3
```

XDebug

XDebug allows you to easily debug php code using modified var_dump() function.
Install **`xdebug`** and uncomment the lines at `/etc/php/conf.d/xdebug.ini`:

```
zend_extension=xdebug
xdebug.remote_enable=on
xdebug.remote_host=127.0.0.1
xdebug.remote_port=9000
xdebug.remote_handler=dbgp
```

Note: To always attempt to start a remote debugging session and connect to a debugging client it is also necessary to set `xdebug.remote_autostart=on`.

IMAP

Install **`php-imap`** and uncomment the line at `/etc/php/php.ini`:

```
extension=imap
```

Caching

There are two kinds of caching in PHP: *opcode*/*bytecode* caching and *userland*/*user data* caching. Both allow for substantial gains in applications speed, and therefore should be enabled wherever possible.

- **Zend OPCache** provides only *opcode* caching.
- **APCu** provides only *userland* caching.

OPCache

OPCache comes bundled with the standard PHP distribution, therefore to enable it you simply have to add or uncomment the following line in your **PHP configuration file**:

```
/etc/php/php.ini
```

```
zend_extension=opcache
```

A list of its options and suggested settings can be found in its **official entry** on the PHP website.

Warning: If you choose to apply the **suggested settings** its manual offers, be sure to read carefully **the first comment** below those instructions as well. In some configurations those settings result in errors such as `zend_mm_heap corrupted` being produced.

APCu

APCu can be installed with the **`php-apcu`** package. You can then enable it by uncommenting the following line in `/etc/php/conf.d/apcu.ini`, or adding it to your **PHP configuration file**:

```
extension=apcu
```

Its author recommends a few **suggested settings**, among which:

- `apc.enabled=1` and `apc.shm_size=32M` are not really required as they represent the **default values**;
- `apc.ttl=7200` on the other hand seems **rather beneficial**;
- finally, `apc.enable_cli=1`, which although **not recommended** by the manual may be required by some software such as **ownCloud**.

Tip: You can add those settings either to APCu's own `/etc/php/conf.d/apcu.ini` **or** directly to your **PHP configuration file**. Just make sure not to enable the extension twice as it will result in errors being diplayed in the system logs.

Development tools

- **Aptana Studio** — IDE for programming in PHP and web development. Does not have a PHP debugger.
  http://www.aptana.com/products/studio3.html || `aptana-studio`AUR

- **Eclipse PDT** — The PHP variant of Eclipse.
  http://www.eclipse.org/pdt/ || `eclipse-php`

- **Komodo** — IDE with good integration for PHP+HTML+JavaScript.
  http://komodoide.com/ || `komodo-ide`AUR, editor only: `komodo-edit`AUR

- **Netbeans** — IDE for many languages including PHP. Includes features like debugging, refactoring, code templating, autocomplete, XML features, and web design and development functionalities.
  https://netbeans.org/ || `netbeans`

- **JetBrains PhpStorm** — Commercial, cross-platform IDE for PHP built on JetBrains' IntelliJ IDEA platform. You can get a free license for education from Jetbrains.[1].
  https://www.jetbrains.com/phpstorm/ || `phpstorm`AUR, 30-day trial: `phpstorm-eap`AUR

- **Zend Studio** — Official PHP IDE, based on eclipse. The IDE has autocomplete, advanced code formatting, WYSIWYG html editor, refactoring, and all the eclipse features such as db access and version control integration and whatever you can get from other eclipse plugins.
  http://www.zend.com/products/studio/ || `zendstudio`AUR

Commandline tools

This article or section needs language, wiki syntax or style improvements.

See Help:Style for reference.

Reason: Use Template:App. (Discuss in Talk:PHP#)

Composer

Composer is a dependency manager for PHP. It can be installed with the `composer` package.

Allow user-wide installations

To allow global package installations for the current **user** (e.g. `$ composer global require "package/name"`), you may want to specify a default location by using an **environment variable**:

```
PATH="$HOME/.config/composer/vendor/bin:$PATH"
```

Box

Box is an application for building and managing Phars. It can be installed with the **php-box**AUR package.

PDepend

PHP Depend (pdepend) is software metrics tool for php. It can be installed with the **pdepend**AUR package.

PHP Coding Standards Fixer

PHP Coding Standards Fixer a is PSR-1 and PSR-2 Coding Standards fixer for your code. It can be installed with the **php-cs-fixer**AUR package.

PHP-CodeSniffer

PHP CodeSniffer tokenizes PHP, JavaScript and CSS files and detects violations of a defined set of coding standards. It can be installed with the **php-codesniffer**AUR package.

phpcov

phpcov is a command-line frontend for the PHP_CodeCoverage library. It can be installed with the **phpcov**AUR package.

phpDox

phpDox is the documentation generator for PHP projects. This includes, but is not limited to, API documentation. It can be installed with the **phpdox**AUR package.

PHPLoc

PHPLoc is a tool for quickly measuring the size of a PHP project. It can be installed with the **phploc**AUR package.

PhpMetrics

PhpMetrics provides various metrics about PHP projects. It can be installed with the **phpmetrics**AUR package.

phptok

phptok is a tool for quickly dumping the tokens of a PHP sourcecode file. It can be installed with the **phptok**AUR package.

PHPUnit

PHPUnit is a programmer-oriented testing framework for PHP. It can be installed with the **phpunit**AUR package.

PHPUnit Skeleton Generator

PHPUnit Skeleton Generator is a tool that can generate skeleton test classes from production code classes and vice versa. It can be installed with the **phpunit-skeleton-generator**AUR package.

Producer

Producer is a command-line quality-assurance tool to validate, and then release, your PHP library package. It can be installed with the **producer**AUR package.

Troubleshooting

PHP Fatal error: Class 'ZipArchive' not found

Ensure the zip extension is enabled.

```
/etc/php/php.ini
---
extension=zip
```

/etc/php/php.ini not parsed

If your `php.ini` is not parsed, the ini file is named after the sapi it is using. For instance, if you are using uwsgi, the file would be called `/etc/php/php-uwsgi.ini`. If you are using cli, it is `/etc/php/php-cli.ini`.

PHP Warning: PHP Startup: <*module*>: Unable to initialize module

When running `php`, this error indicates that the aforementioned module is out of date. This will rarely happen in Arch Linux, since maintainers make sure core PHP and all modules be only available in compatible versions.

This might happen in conjunction with a module compiled from the **AUR**. You usually could confirm this by looking at the dates of the files `/usr/lib/php/modules/`.

To fix, find a compatible update for your module, probably by looking up the **AUR** using its common name.

If it applies, flag the outdated **AUR** package as *outdated*.

See also

- **PHP Official Website**

Category:

- Programming languages