<?php

Setting Up PHP

Mike Naberezny
Senior PHP Developer
mike.n@zend.com

Matthew Weier O'Phinney
PHP Developer
matthew@zend.com

Zend The php Company

Setting Up PHP

Topics

- Where to get PHP
- Linux Installation
- Apache Installation
- Windows Considerations
- PHP.INI Settings
- PEAR and PECL
- Developing with PHP
- Security Concerns

Where to Get PHP

Resources

- http://php.net/downloads: Default download area. Get the latest stable PHP5 and PHP4 releases, and often the most recent release candidate of PHP5.

- http://snaps.php.net: Snapshots, updated daily.

- http://cvs.php.net: Anonymous checkout from the public CVS Repository.

Compilation on Linux

Compilation on Linux

Compilation on Linux

Why Compile?

- Distros are usually one or more releases behind current stable
- Ability to compile in only those modules you need; tune PHP for your applications
- Ability to compile multiple versions of PHP for testing

Compilation on Linux

Compilation Flags

- Order and positioning of flags occasionally matters
- Use --prefix to specify a unique location for this version of PHP

Compilation on Linux

SAPI Flags

- --with-apxs/--with-apxs2: Select Apache1 or Apache2
- --disable-cli: Optionally disable the CLI SAPI
- --disable-cgi: Optionally disable the CGI SAPI
- --enable-fast-cgi: Optionally enable Fast CGI SAPI
- --disable-path-info-check: Optionally disable PATH_INFO

Compilation on Linux

General Settings

- --with-config-file-path: specify a custom path to php.ini
- --enable-safe-mode (don't!)
- --enable-magic-quotes (don't!)
- --disable-short-tags (maybe)

Compilation on Linux

Extensions

- --disable-all: Only for the experts
- --with-zlib: Used for a variety of extensions, including output compression, jpeg support, and more.
- --disable-ctype: don't. These are lightning fast functions for testing input types.
- --with-curl/--with-curlwrappers: Alternate to allow_url_fopen(), only more powerful.

Compilation on Linux

Extensions (cont.)

- --with-dom (PHP4) / --disable-dom (PHP5): include (or remove) DOM support for XML and HTML.
- --enable-exif: Analyze image metadata.
- --enable-ftp: Open FTP files via streams.
- --with-gd: Create, manipulate, and analyze images. GIF support is built-in; you'll need libjpeg, libpng, and/or libXpm for JPEG, PNG, and XPM support (and zlib for PNG support); you may also want freetype to create graphical text.

Compilation on Linux

Extensions (cont.)

- --with-gettext: Add NLS to your applications. PHP can only read translation files; use gettext to create them.
- --with-imap: IMAPv4, POP3, and NNTP support.
- --with-ldap: Authenticate against LDAP directories
- --enable-mbstring/--enable-mbregex: Allow multibyte strings and multibyte-enabled regexes. Useful when dealing with multibyte character sets.

Compilation on Linux

Extensions (cont.)

- --with-mcrypt/--with-mhash: Encrypt and decrypt values, or create checksums and file signatures. PEAR 1.4.x utilizes functionality from these libraries, which unfortunately are not standard on many distros.
- --with-mysql/--with-mysqli: MySQL support. MySQLi takes advantage of special features of MySQL 4.1 and above, greatly enhancing performance.
- --with-custom-odbc/--with-iodbc/--with-unixODBC: Connect to ODBC data sources.

Compilation on Linux

Zend
The *php* Company

Extensions (cont.)

- --with-ncurses: Create pretty CLI interfaces.
- --with-readline: Provide tab completion and history to CLI interfaces. In PHP5, you can alternately use --with-libedit.
- --with-pspell: Spell check from PHP.
- --disable-overload (PHP4): Disable user-space object overloading (__get, __set, and __call).
- --with-pdflib: Create PDFs from PHP. Requires a license.

Compilation on Linux

Extensions (cont.)

- --enable-pcntl: Allow process control, such as forking, from PHP.
- --without-pcre-regex/--disable-posix: Disable PCRE or POSIX regexes.
- --disable-session: Disable PHP session support.
- --with-msession: Enable Mohawk Session server support.
- --with-mm: Enable memory based sessions.
- --enable-shmop: Enable shared memory segments.

Compilation on Linux

Extensions (cont.)

- --disable-simplexml (PHP5): Disable PHP5's SimpleXML support.
- --disable-xml: Disable XML support -- which will also disable XMLRPC, SOAP, SimpleXML, and DOM support.
- --with-xmlrpc: Enable PHP native XMLRPC support.
- --with-xsl: Use XSL to transform XML within PHP.
- --enable-soap (PHP5): Enable SOAP support.
- --enable-sockets: Enable socket support for streams.

Compilation on Linux

Extensions (cont.)

- --disable-spl (PHP5): Disable the Standard PHP Library. This will disable features such as Iterators.
- --with-tidy (PHP5): Enable the tidy extension.
- --disable-tokenizer (PHP4): Disables HTML tokenizer -- which, if present, greatly speeds processing in phpDocumentor.

Compilation in Linux

PEAR flags

- --disable-cli also disables the ability to install PEAR, which needs the CLI SAPI during installation.
- --with-pear=<dir>: Specify the location for your PEAR installation.
- --without-pear: Don't install PEAR.

Compilation on Linux

Install php.ini

- If --with-config-file-path was used, look for it there.
- Otherwise
  - Copy php.ini-dist and php.ini-recommended to $PREFIX/lib
  - php.ini-dist: Usually safe for developers.
  - php.ini-recommended: Recommended settings for production servers.

Apache Installation

Apache Installation

Three Basic Choices:

- As an Apache module (mod_php)
  - Use --with-apxs or --with-apxs2 flags
- As a vanilla CGI module
- As a Fast CGI module
  - Use --enable-fast-cgi

Apache Installation

PHP Directives Available

- php_value
- php_admin_value

Apache Installation

Multiple Simultaneous PHP Versions

- Two methods:
  - Use multiple Apache processes, listening on different ports
  - Use multiple Apache processes, with a gateway process running mod_proxy

Apache Installation

Multiple Simultaneous PHP Installations

- Multiple Ports method
  - Each PHP version has a corresponding httpd.conf that loads its module and listens on a different port.
  - Start each apache process with 'httpd -c /path/to/httpd.conf'
  - Browse to servername:port

Apache Installation

Multiple Simultaneous PHP Installations

- mod_proxy method
  - Compile apache with --enable-module=proxy
  - Have multiple httpd.conf files:
    - Gateway process uses vhosts to proxy to the others
    - Other processes listen on localhost on different ports and load the desired PHP module.

Setting Up PHP

PHP.INI Settings

PHP.INI Settings

Error Reporting

- Development Servers
  - log_errors=Off
  - display_errors=On
  - error_reporting
    - PHP4: E_ALL
    - PHP5: E_ALL | E_STRICT
  - track_errors=On

PHP.INI Settings

Error Reporting (cont.)

- Production Servers
  - log_errors=On
  - error_log=/path/to/error.log
  - display_errors=Off
  - error_reporting E_ALL & ~E_NOTICE
  - Build an error handler that will email you. You can place this in an auto_prepend_file or run it as a cron job.

PHP.INI Settings

- short_tags/asp_tags: ASP tags should always be off, as they're not portable. short_tags (<? ?>, <?= ?>) are used in many legacy apps, and may need to be on.
- register_globals = Off
- register_long_arrays = Off
- magic_quotes_gpc/magic_quotes_runtime = Off
- register_argc_argv: Useful for CLI scripts. However, it's a slight performance hit when you're only running as CGI or an Apache module.

PHP.INI Settings

- disable_functions: Comma separated list of PHP internal functions to disable.
- disable_classes: Comma separated list of PHP internal classes to disable.
- expose_php: Turn it off for a very slight performance gain, and to obscure what languages you're running on your server.

PHP.INI Settings

File Upload Directives

- Only turn on file_uploads if your apps need it.
- post_max_size: Can be set in php.ini, httpd.conf, or .htaccess. Limits size of entire POST request, which could contain multiple file uploads.
- upload_max_filesize: php.ini, httpd.conf, and .htaccess; limit the size of any one file upload.
- upload_tmp_dir: Indicate where upload files should be placed on the server.

PHP.INI Settings

- max_execution_time: Set a reasonable default -- the standard default is 30 seconds, which may be longer than most processes. You can override this at any level, including individual scripts.
- allow_url_fopen: This can sometimes be a security hole if fopen() is used with user input. You can replace it with the cURL functions.

PHP.INI Settings

Session Directives

- session.save_handler: 'files' is the only value. You override this with the session_set_save_handler() function, which you may use to save sessions to a persistent store, such as a database.
- session.use_trans_sid: enable propagation of session Ids in the URL and forms. Not recommended.
- session.use_cookies: turn on to force usage of cookies for session identifiers.
- session.save_path: can include a level and mode: “2;600;/path” saves sessions in a 2-tier hierarchy.

Session Directives (cont.)

- session.name: set a default name for the session identifier.
- session.auto_start = 0; you should only start sessions when you need them, as there is overhead involved with them.
- session.cookie_lifetime / session.cookie_path / session.cookie_domain: Set the default cookie information for the session cookie.

PHP.INI Settings

Session Directives (cont.)

- session.gc*: finetune the session garbage collection.
  - session.gc_probability
  - session.gc_divisor
  - Probability of garbage collection = gc_probability/gc_divisor
  - session.gc_maxlifetime: indicates at what point after last modification time a session is considered garbage so it will be cleaned up by the garbage collection process. Industry standard is 24 minutes.
- Alternately, use a cronjob for garbage collection.

PHP.INI Settings

Mail Settings

- *nix: sendmail_path
  - Can be an executable and arguments. Does not necessarily need to be sendmail -- you can set up a PHP script that uses SMTP, use an agent such as MSMTP, etc. However, the script needs to be able to process a standard email piece.
- Windows: SMTP and smtp_port
  - Specify an SMTP server (and port, if necessary) via which to send an email

PHP.INI Settings

- include_path: Specify a colon-delimited (*nix) or semicolon-delimited (Windows) list of paths that should be searched when an include() or require() is performed. Typical usage is to place common libraries outside the document root of the web server.
- open_basedir: Like include_path. However, this limits any include() or require() statements to the paths (and subtrees) specified. This can be set in php.ini, or also in httpd.conf (useful for virtual hosts).

PEAR and PECL

PEAR and PECL

PEAR:

PHP Extension and Application Repository

http://pear.php.net

Provides:

- Library of common and tested code
- Installer for PHP code
- Distribution of PHP code (via custom channels)

PEAR and PECL

Installation

- go-pear (*nix) or go-pear.bat (Windows)
- http://go-pear.org

PEAR and PECL

Usage

- pear help: lists all commands
- pear help command: lists options for a specific command
- pear install --alldeps package: installs a package, and fetches and installs all dependencies.
  - Use packagename-alpha, packagename-beta, packagename-devel to get non-stable packages.
- pear upgrade --alldeps package / pear upgrade-all
- pear info package: get information on a package.

PEAR and PECL

Useful PEAR Packages

- Cache_Lite: fast caching of script output or blocks.
- Log: log messages to memory, mail, files, or databases.
- DB and MDB2: Database abstraction.
- Pager: pagination navigation.
- HTTP_Upload: simplified upload handling.
- Session: pre-configured session handlers.
- PhpDocumentor: the standard for PHP documentation.
- PhpUnit: powerful unit test library.

PEAR and PECL

PECL:
PHP Extension Community Library

http://pecl.php.net/

http://pecl4win.php.net/ (Windows Ports)

Provides:

- Non-standard C extensions to the PHP engine.

PEAR and PECL

Usage

- Same as for pear. You can use pear to install pecl packages, or the command 'pecl' (which is an alias to the pear command).
- May not work with PHP distributed by some Linux distributions (a reason to compile your own PHP).

PEAR and PECL

Useful PECL Packages

- classkit: runtime definition and modification of classes and methods.
- imagick: use ImageMagick via PHP (without resorting to the shell).
- bz2: build and access bz2 compressed files.
- tidy: clean and format X/HTML output from your scripts.
- svn: subversion access for PHP.
- xdebug: PHP debugging extension.
- PDO: PHP data objects.

Developing with PHP

Useful Tools

- PhpDocumentor (http://www.phpdoc.org/)
  - Document your code in the phpdoc format, and gain API documentation for your projects. Documentation can be at the variable, function, class, method, and package levels.

Useful Tools

- PhpUnit (http://www.phpunit.de/)
  - Unit Testing framework for PHP based on JUnit. PHPUnit2 is PHP5-only.

Security Concerns

Security Concerns

File Permissions

- Typically, scripts should be read-only.
- If your script writes data to the filesystem, have a separate directory that's owned by the process running the web server, and writeable only by that user; store all script-generated (or uploaded) data in that directory tree.

Security Concerns

INI Settings to Watch

- safe_mode
- open_basedir
- magic_quotes_gpc / magic_quotes_runtime
- register_globals
- session.autostart
- allow_url_fopen
- upload_max_filesize

Security Concerns

Password Storage

- Have a function stored in a file outside the document root return the credentials.
- Use SetEnv in httpd.conf to create server environment variables containing the credentials:
    - SetEnv “MYSQL_USER” “matthew”
    - These values can be echoed via phpinfo(), posing a slight security risk.

Questions?