**PHP 5.x : MySQL**

Personal Home Page (1995)

PHP: Hypertext Preprocessor

PHP History

# Development

PHP started out as a small open source project that evolved as more and more people found out how useful it was. **Rasmus Lerdorf** unleashed the first version of PHP way back in 1994.

## Rasmus Lerdorf

Rasmus Lerdorf first started developing PHP/FI. He could not have imagined that his creation would eventually lead to the development of PHP as we know it today, which is being used by millions of people. The first version of “PHP/FI,” called Personal Homepage Tools/Form Interpreter, was a collection of Perl scripts in 1995. One of the basic features was a Perl-like language for handling form submissions, but it lacked many common useful language features, such as for loops.

### PHP Introduction

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor". “Personal Home Page”
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

### 

### Common uses of PHP

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
* You add, delete, modify elements within your database through PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

### Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

**What is Php ?**

* PHP stands for PHP: Hypertext Preprocessor
* PHP is a server-side scripting language, like ASP
* PHP scripts are executed on the server
* PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid,
* PostgreSQL, Generic ODBC, etc.)
* PHP is an open source software
* PHP is free to download and use

**What is a PHP File?**

* PHP files can contain text, HTML tags and scripts
* PHP files are returned to the browser as plain HTML
* PHP files have a file extension of ".php", ".php3", or ".phtml"

**Why PHP?**

* PHP runs on different platforms (Windows, Linux, Unix, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP is FREE to download from the official PHP resource: [www.php.net](http://www.php.net)
* PHP is easy to learn and runs efficiently on the server side

**Explain Features of PHP.**

* PHP runs on different platforms (Windows, Linux, Unix, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP is FREE to download from the official PHP resource: [www.php.net](http://www.php.net)
* PHP is easy to learn and runs efficiently on the server side

**There are three main fields where PHP scripts are used.**

1. Server side scripting
2. Command line scripting.
3. Writing client-side GUI applications.For this PHP-GTK is used.

PHP-GTK is an extension for the PHP programming language that implements language bindings for GTK+. It provides an object-oriented interface to GTK+ classes and functions and greatly simplifies writing client-side cross-platform GUI applications.

* PHP can be used on all major operating systems, including Linux, many Unix
* variants, Microsoft Windows, Mac OS X etc.
* PHP has also support for most of the web servers today. This includes Apache,
* Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet
* servers, Oreilly Website Pro server and many others.
* You also have the choice of using procedural programming or object oriented programming, or a mixture of them.
* PHP does more than just generating dynamic web-pages.
* PHP's abilities includes:
  + Generating images dynamically
  + PDF files
  + Flash movies
  + Compression
  + Download and upload
  + XML support
* PHP also has support for talking to other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (on Windows) and countless others.
* You can also open raw network sockets and interact using any other protocol.
* PHP has support for the WDDX complex data exchange between virtually all Web
* programming languages. (Support for web services)
* PHP has support for instantiation of Java objects and using them transparently as PHP
* objects. You can also use CORBA extension to access remote objects.
  + e.g. You can use java classes in php.

PHP Tools



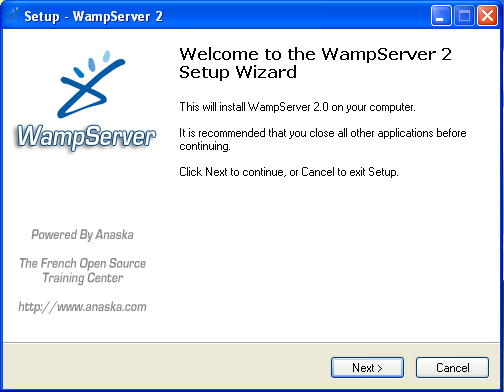
* **Windows OS Windows Apache MySQL PHP (WAMP)**
* **Linux OS Linux Apache MySQL PHP (LAMP)**
* **Mac OS MacOS Apache MySQL PHP (MAMP / MAMP Pro)**
* **All OS Cross-Platform Apache MySQL PHP/Perl (XAMPP)**

### Browsers

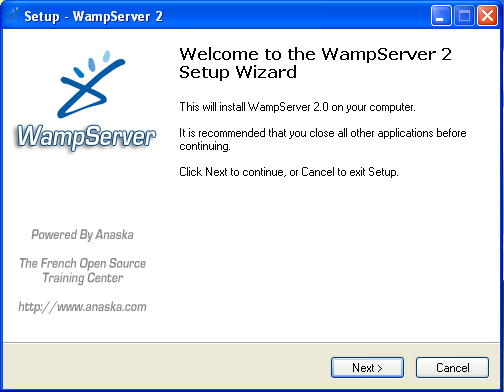


WAMP Installation

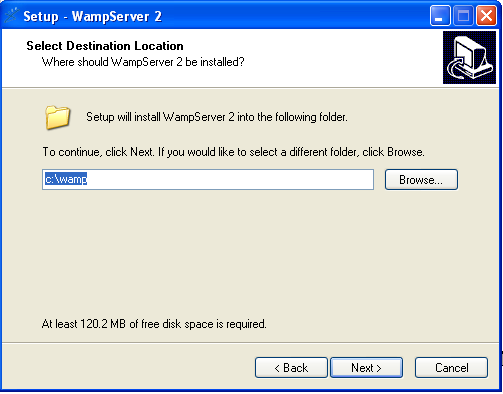
1. Double Click on WAMP Setup then Following Dialogue box is open



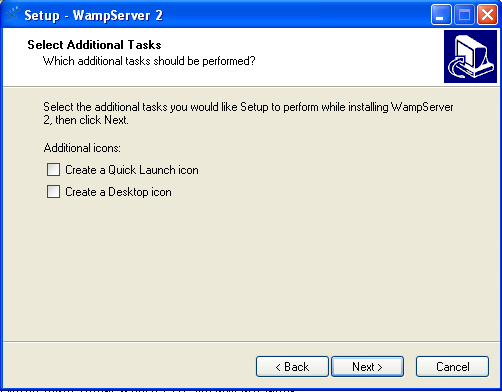
1. Please Click on Next then following dialogue box is open. In that Click on Click on I accept the agreement.



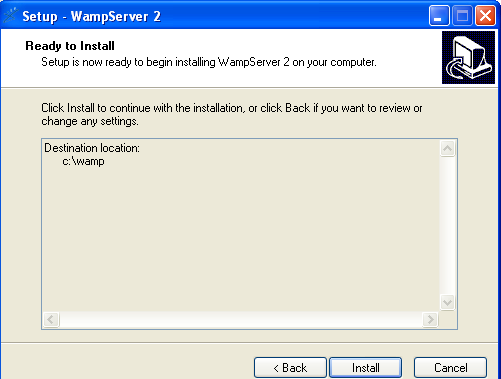
1. To Continue click next .If you would like to select a different folder. Click browse



1. After Click next Following Dialogue box is open



1. Click Install to continue installation.



### PHP IIS Server

Follow the instruction from this URL

<https://www.howtogeek.com/50432/how-to-install-php-on-iis-7-for-windows-server-2008/>

Or follow this video tutorials

<https://www.youtube.com/watch?v=3Q27FJYmpwQ>

#### PHP.ini file

The PHP configuration file, php.ini, is the final and most immediate way to affect PHP's functionality. The php.ini file is read each time PHP is initialized.in other words, whenever httpd is restarted for the module version or with each script execution for the CGI version. If your change isn't showing up, remember to stop and restart httpd. If it still isn't showing up, use phpinfo() to check the path to php.ini.

The configuration file is well commented and thorough. Keys are case sensitive, keyword values are not; whitespace, and lines beginning with semicolons are ignored. Booleans can be represented by 1/0, Yes/No, On/Off, or True/False. The default values in php.ini-dist will result in a reasonable PHP installation that can be tweaked later.

Here we are explaining the important settings in php.ini which you may need for your PHP Parser.

### short\_open\_tag = Off

Short open tags look like this: <? ?>. This option must be set to Off if you want to use XML functions.

### safe\_mode = Off

If this is set to On, you probably compiled PHP with the --enable-safe-mode flag. Safe mode is most relevant to CGI use. See the explanation in the section "CGI compile-time options". earlier in this chapter.

### safe\_mode\_exec\_dir = [DIR]

This option is relevant only if safe mode is on; it can also be set with the --with-exec-dir flag during the Unix build process. PHP in safe mode only executes external binaries out of this directory. The default is /usr/local/bin. This has nothing to do with serving up a normal PHP/HTML Web page.

### safe\_mode\_allowed\_env\_vars = [PHP\_]

This option sets which environment variables users can change in safe mode. The default is only those variables prepended with "PHP\_". If this directive is empty, most variables are alterable.

### safe\_mode\_protected\_env\_vars = [LD\_LIBRARY\_PATH]

This option sets which environment variables users can't change in safe mode, even if safe\_mode\_allowed\_env\_vars is set permissively

### disable\_functions = [function1, function2...]

A welcome addition to PHP4 configuration and one perpetuated in PHP5 is the ability to disable selected functions for security reasons. Previously, this necessitated hand-editing the C code from which PHP was made. Filesystem, system, and network functions should probably be the first to go because allowing the capability to write files and alter the system over HTTP is never such a safe idea.

### max\_execution\_time = 30

The function set\_time\_limit() won’t work in safe mode, so this is the main way to make a script timeout in safe mode. In Windows, you have to abort based on maximum memory consumed rather than time. You can also use the Apache timeout setting to timeout if you use Apache, but that will apply to non-PHP files on the site too.

### error\_reporting = E\_ALL & ~E\_NOTICE

The default value is E\_ALL & ~E\_NOTICE, all errors except notices. Development servers should be set to at least the default; only production servers should even consider a lesser value

### error\_prepend\_string = [""]

With its bookend, error\_append\_string, this setting allows you to make error messages a different color than other text, or what have you.

### warn\_plus\_overloading = Off

This setting issues a warning if the + operator is used with strings, as in a form value.

### variables\_order = EGPCS

This configuration setting supersedes gpc\_order. Both are now deprecated along with register\_globals. It sets the order of the different variables: Environment, GET, POST, COOKIE, and SERVER (aka Built-in).You can change this order around. Variables will be overwritten successively in left-to-right order, with the rightmost one winning the hand every time. This means if you left the default setting and happened to use the same name for an environment variable, a POST variable, and a COOKIE variable, the COOKIE variable would own that name at the end of the process. In real life, this doesn't happen much.

### register\_globals = Off

This setting allows you to decide whether you wish to register EGPCS variables as global. This is now deprecated, and as of PHP4.2, this flag is set to Off by default. Use superglobal arrays instead. All the major code listings in this book use superglobal arrays.

### magic\_quotes\_gpc = On

This setting escapes quotes in incoming GET/POST/COOKIE data. If you use a lot of forms which possibly submit to themselves or other forms and display form values, you may need to set this directive to On or prepare to use addslashes() on string-type data.

### magic\_quotes\_runtime = Off

This setting escapes quotes in incoming database and text strings. Remember that SQL adds slashes to single quotes and apostrophes when storing strings and does not strip them off when returning them. If this setting is Off, you will need to use stripslashes() when outputting any type of string data from a SQL database. If magic\_quotes\_sybase is set to On, this must be Off.

### magic\_quotes\_sybase = Off

This setting escapes single quotes in incoming database and text strings with Sybase-style single quotes rather than backslashes. If magic\_quotes\_runtime is set to On, this must be Off.

### auto-prepend-file = [path/to/file]

If a path is specified here, PHP must automatically include() it at the beginning of every PHP file. Include path restrictions do apply.

### auto-append-file = [path/to/file]

If a path is specified here, PHP must automatically include() it at the end of every PHP file.unless you escape by using the exit() function. Include path restrictions do apply.

### include\_path = [DIR]

If you set this value, you will only be allowed to include or require files from these directories. The include directory is generally under your document root; this is mandatory if you.re running in safe mode. Set this to . in order to include files from the same directory your script is in. Multiple directories are separated by colons: .:/usr/local/apache/htdocs:/usr/local/lib.

### doc\_root = [DIR]

If you.re using Apache, you.ve already set a document root for this server or virtual host in httpd.conf. Set this value here if you.re using safe mode or if you want to enable PHP only on a portion of your site (for example, only in one subdirectory of your Web root).

### file\_uploads = [on/off]

Turn on this flag if you will upload files using PHP script.

### upload\_tmp\_dir = [DIR]

Do not uncomment this line unless you understand the implications of HTTP uploads!

### session.save-handler = files

Except in rare circumstances, you will not want to change this setting. So don't touch it.

### ignore\_user\_abort = [On/Off]

This setting controls what happens if a site visitor clicks the browser.s Stop button. The default is On, which means that the script continues to run to completion or timeout. If the setting is changed to Off, the script will abort. This setting only works in module mode, not CGI.

### mysql.default\_host = hostname

The default server host to use when connecting to the database server if no other host is specified.

### mysql.default\_user = username

The default user name to use when connecting to the database server if no other name is specified.

### mysql.default\_password = password

The default password to use when connecting to the database server if no other password is specified.