PHP.INI File Configuration

On installing PHP software on your machine, php.ini is created in the installation directory. In case of XAMPP, php.ini is found in c:\xamm\php folder. It is an important configuration file that controls the performance and sets all the PHP related parameters.

The phpinfo() function displays a list of different parameters and their current values of PHP, Aache, MySQL and other parts of the web server installation.

Run the following code to display the settings, one of which shows the path to the "php.ini" file:

```
<?php
   echo phpinfo();
?>
```

Loaded Configuration File

Locate the Loaded Configuration File setting that displays the location of php.ini file

```
C:\xampp\php\php.ini
```

Different aspects of PHP's behaviour are configured by a large number of parameters (called directives). The "php.ini" file comes with most of the lines starting with semicolon (;) symbol – indicating that the line is commented. The uncommented line is actually the effective directive and its value. In other words, to activate and assign a value to a particular directive, remove the leading semicolon.

```
directive = value
```

Directive names are *case sensitive. Directives are variables used to configure PHP or PHP extensions. Note that there is no name validation, so if an expected directive is not found a default value will be used, which can be a string, a number, a PHP constant (e.g. E_ALL or M_PI), one of the INI constants (On, Off, True, False, Yes, No and None).

Actually, the C:\XAMPP\PHP folder contains two INI files, one to be used in production environment and other in development environment.

The php.ini-development.ini is very similar to its production variant, except it is much more verbose when it comes to errors. In development stage, copy this as php.ini to be able to trace the bugs in the code. Once the code is ready for deployment, use php.ini-production.ini file as the effective php.ini file, which essentially supress the error messages to a large extent.

The directives in php.ini are divided in different categories, like Error handling, data handling, path and directories, file uploads, PHP extensions and module settings.

Here is a list of some of the important directives in "php.ini" file:

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 time out 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 are using Apache, you have 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 comment 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.