

Installing PHP 5.1 On Windows

1. The PHP setup file is a zip archive. Extract the contents of the zip file using Winzip or any other archiving tool
2. In the extraction process, all the zipped files will be extracted into a folder, whose name is based on the version of PHP that was downloaded. For example, if **php-5.1.0RC1-Win32.zip** was downloaded and extracted to **C:**, there will be a folder called **C:\php-5.1.0RC1-Win32** where the files extracted can be found
3. Rename the folder **php-5.1.0RC1-Win32** to **php** (Refer to diagram 1.1)

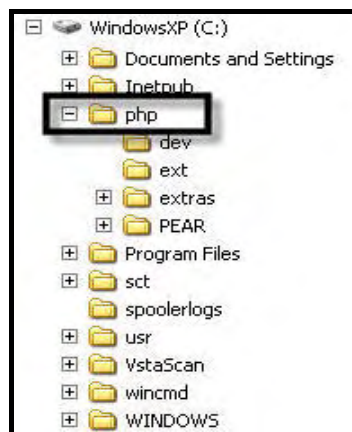


Diagram 1.1: The Directory Structure

4. To integrate **PHP** with **Apache2** there are three options available:
 - a. Copy the file named **php5ts.dll** available under **c:\php** to the Windows system directory
 - b. Copy the file named **php5ts.dll** to the web server's directory
 - c. Add the PHP directory i.e. **c:\php** to the PATH variable

The use of the third option is recommended, as this option will help when upgrading to a newer version of PHP in future.

Add **c:\php** to the PATH variable as follows:

- a. Select **Start** → **Control Panel** → **System**. **System Properties** dialog box appears as shown in diagram 1.2

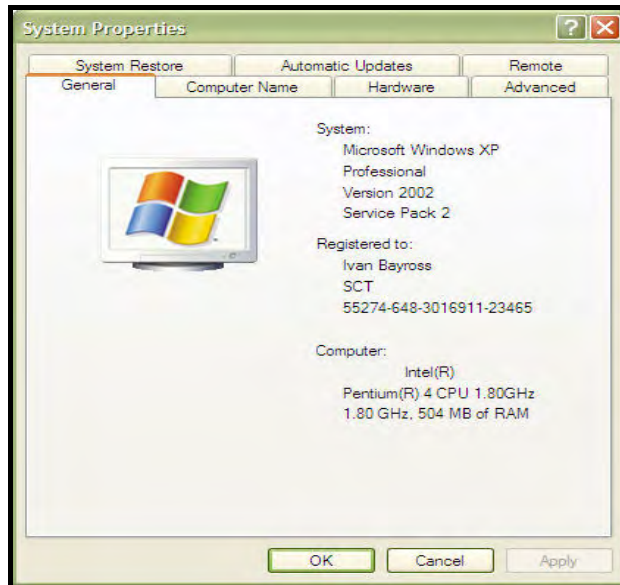


Diagram 1.2: The System Properties in control panel

- b. Select the **Advanced** tab. Refer to diagram 1.3

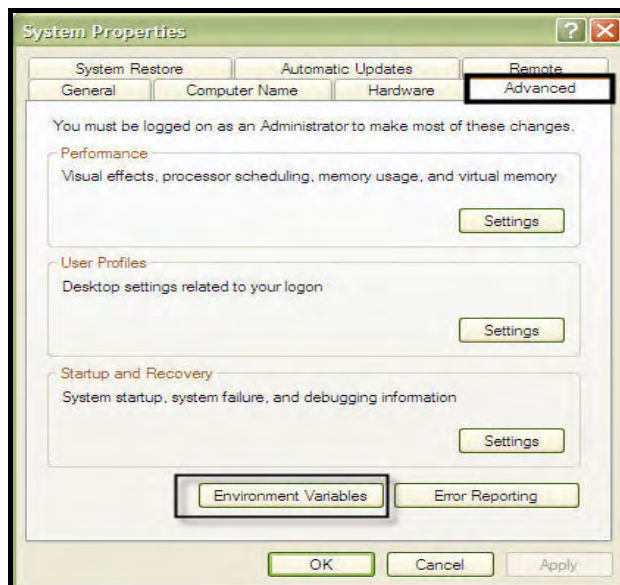


Diagram 1.3: The System Properties Advanced Tab

INSTALLING PHP 5.1 ON WINDOWS

- c. Click **Environment Variables**. This pops up a window as shown in diagram 1.4

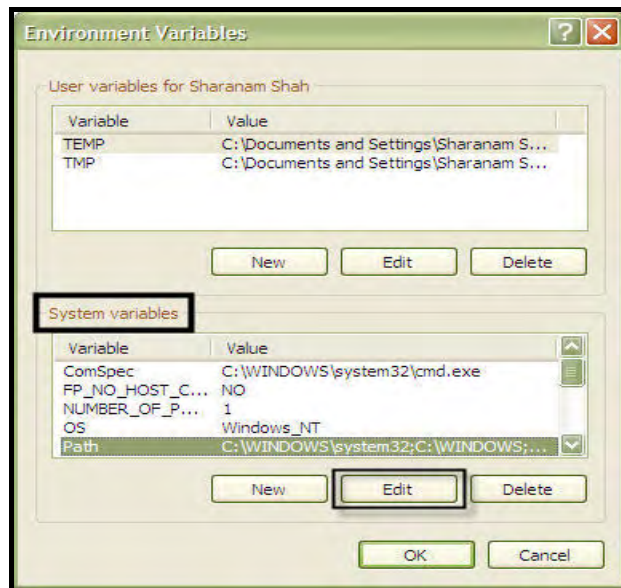


Diagram 1.4: The System Properties Environment Variables

- d. Select the **Path** variable under **System variables** section and click **Edit**. This pops up a window, in the text box **Variable value** add **;C:\php** at the end of the path as shown in diagram 1.5

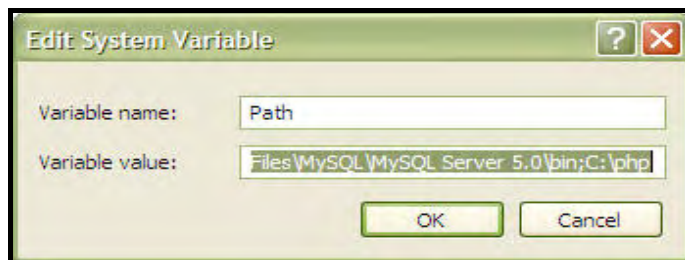


Diagram 1.5: The Edit System variable window

- e. Click **OK** to apply these settings
- 5. The next step is to set up a valid configuration file for PHP i.e. **php.ini**. There are two **ini files** distributed in the zip file i.e. **php.ini-dist** and **php.ini-recommended**. Use the **php.ini-recommended**, as it is optimized with the default settings for performance and security

6. Copy the chosen ini-file (i.e. **php.ini-recommended**) to the windows directory (i.e. c:\windows **or** c:\winnt as the case may be) directory and rename it to **php.ini**



If the file system in use is NTFS on Windows NT, 2000, XP or 2003, make sure that the user running the web server has read permissions to the **php.ini**.

PHP is now setup on the Windows system.

Binding The PHP Installation With Apache2

Apache does not know that PHP is just installed. Therefore Apache needs to be informed about PHP especially where to find it.

This is done via Apache's httpd.conf file. Apache reads this file and understands what modules need to be loaded and where these modules are located.



It is not mandatory to have Apache web server installed in order to test .php scripts. These scripts can simply be run using the interpreter **php.exe**. This can be accomplished by appending the .php script file as a command line argument to **php.exe** interpreter.

There are **two ways** to configure Apache to use PHP.

One is to configure it to load the PHP interpreter as an Apache module.

The other is to configure it to run the PHP interpreter as a CGI binary.



It is recommended that PHP is loaded as a module in Apache, since it runs more efficiently that way, unless there is a specific reason for running PHP as a CGI binary.

Edit Apache's **httpd.conf** file. Make sure the **PHP mime type** is specified and **uncommented**. The line should look like this: (Refer to diagram 1.6)

```
AddType application/x-httpd-php .php
```

This line means that every file that ends with .php will be processed as a PHP file.

INSTALLING PHP 5.1 ON WINDOWS



If need arises to support other file types, like .php3, .html, .asp and .phtml, simply add them to the list, like this:

```
AddType application/x-httpd-php .php3
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php .html
AddType application/x-httpd-php .asp
```



The **httpd.conf** file is usually available under **C:\Program Files\Apache Group\Apache2\conf** on Windows.

Ensure that the PHP module is loaded by adding the following line in the httpd.conf file

```
LoadModule php5_module "c:/php/php5apache2.dll"
```

This line tells Apache from where to load the **.dll** file in case of Windows or the **.so** file in case of Linux which is required to execute PHP. This line enables loading the PHP module dynamically into Apache. Usually in Linux the PHP source installer automatically inserts this line. If this line does not exist then insert it manually in the httpd.conf file.

```
httpd.conf - Notepad
File Edit Format View Help
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
#Addition For PHP 5.1 integration
LoadModule php5_module "c:/php/php5apache2.dll"
AddType application/x-httpd-php .php
LoadModule access_module modules/mod_access.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_module modules/mod_auth.so
#LoadModule auth_anon_module modules/mod_auth_anon.so
#LoadModule auth_dbm_module modules/mod_auth_dbm.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule dir_module modules/mod_dir.so
```

Diagram 1.6: The **httpd.conf** file modification



In the Apache 1.3.X, **AddModule** directive was used instead of **LoadModule**.

In case of Apache 1.3.X, search for the block AddModule statement. Add the following line after the last AddModule statement:

AddModule mod_php5.c

mod_php5.c file is not available anywhere in the file system. It only specifies the order in which the Apache Web server enables the various modules.

If Apache 2.X is used, **then do not insert** the AddModule directive. It's no longer needed in that version. Apache 2.X has its own internal method of determining the correct order of loading the modules.

Restart Apache server. PHP files should be able to be served up now.

Testing PHP / Apache

Registering Changes Made In The httpd.conf of Apache2

After making any changes to the **httpd.conf** ensure that Apache is restarted to register the new changes with Apache. Do this by using the icon on the task bar: (Refer to diagram 1.7)

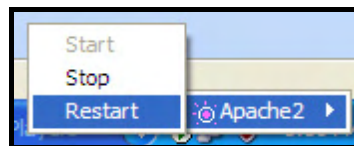


Diagram 1.7: Restarting Apache2

To test whether PHP has been successfully setup and integrated with Apache2 create a simple script named phpinfo.php that contains the following code: (Refer to diagram 1.8)

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

INSTALLING PHP 5.1 ON WINDOWS

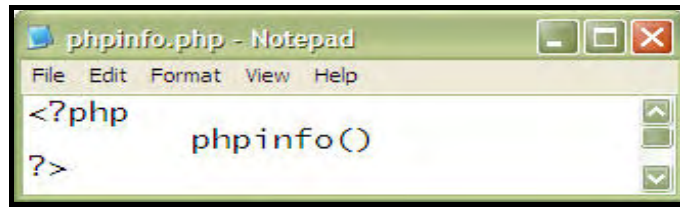


Diagram 1.8: The phpinfophp file in Notepad

Place this file in Apache's default document root directory i.e. **C:\Program Files\Apache Group\Apache2\htdocs**. Examine the output of this script in a Web browser by pointing to **http://127.0.0.1/phpinfo.php**. If PHP setup is successful then a screen similar to that shown in diagram 1.9 will be displayed.

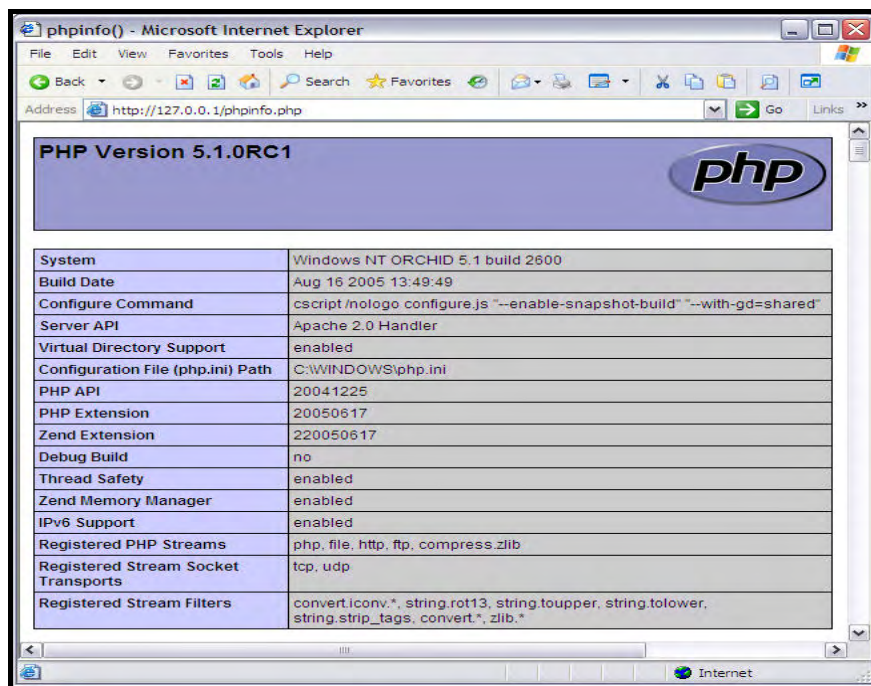


Diagram 1.9: PHP Version Info displayed via Web Browser