**Installing PHP.**

The path used for PHP is just an example, you can choose another if you want.

Extract the archive in C:\PHP (Rename it if necessary)

Rename C:\PHP\php.ini-recommended to C:\PHP\PHP.INI

**Configure the Session directory**

Open C:\PHP\PHP.INI

**Make sure you remove the initial semicolons!**

Find

;session.save\_path = "/tmp"

replace it with

session.save\_path = "C:\WINDOWS\TEMP"

Find

; \*\*You CAN safely turn this off for IIS, in fact, you MUST.\*\*

; cgi.force\_redirect = 1

replace it with

; \*\*You CAN safely turn this off for IIS, in fact, you MUST.\*\*

cgi.force\_redirect = 0

**Configure PHP extensions**

Find

; Directory in which the loadable extensions (modules) reside.

extension\_dir = "./"

replace it with

; Directory in which the loadable extensions (modules) reside.

extension\_dir = "C:\PHP\EXT"

**If you can't find "extension\_dir" in your C:\PHP\PHP.INI file, add it to the bottom of the file.**

**MySQL extension**

As Gallery 2 uses a database to store it's metadata, you need to enable database support in PHP. This guide uses MySQL, but the procedure would be similar for Postgres or Oracle.

Find

;extension=php\_mysql.dll

replace it with

extension=php\_mysql.dll

**Gettext extension**

In order to make the localization of g2 (multi-language) work you need the gettext extension of php. This can be enabled in php.ini. G2 does hint you for that. However gettext is a little strange extension.

Find

;extension=php\_gettext.dll

replace it with

extension=php\_gettext.dll

But now comes the crux. php\_gettext.dll is depending on \php-install-dir\dll\iconv.dll All other extensions work flawlessly for me. But gettext.dll required me to put iconv.dll into a dir that is included in the searchpath. E.g. /windows/system32 I then overreacted and copied all dll's to that /system32 dir. The manual of php 4 tells you to copy the dll's to the /php-install-dir/ but that only works if you add manually the php dir into the path statement of windows.

**GD2 extension**

Find the extension in your php.ini and remove the # in front of the line ;extension=php\_gd2.dll

Find

;extension=php\_gd2.dll

replace it with

extension=php\_gd2.dll

**PHP.INI key values**

Changing key php.ini values to suit the bigger demands of G2: The following values should be working for most of the gallery users. The max times have been upped just as a precaution. But I think that G2 overrides them anyway.

Find the following values and set to your needs.

max\_execution\_time = 300 ; Maximum execution time of each script, in seconds

max\_input\_time = 300 ; Maximum amount of time each script may spend parsing request data

memory\_limit = 32M ; Maximum amount of memory a script may consume (8MB)

upload\_tmp\_dir = "C:\Inetpub\wwwroot\uploads" ; Temporary directory for HTTP uploaded files.

post\_max\_size = 7M ; Maximum size of POST data that PHP will accept.

upload\_max\_filesize = 6M ; The maximum allowed size for uploaded files.

open\_basedir = [path you want to use]. See: [[1]](http://codex.gallery2.org/Gallery2:Security#Restricting_Read_.26_Write_Access_to_Gallery_2_Files_and_Directories)

Set **upload\_max\_filesize** and **post\_max\_size** to whatever you need, but **post\_max\_size** MUST be >= the value you set for **upload\_max\_filesize**, and must be < the value in **memory\_limit**.

**Make PHP available to IIS**

**Set the system path to include C:\PHP**

Click on My Computer -> Properties -> Advanced -> Environment Variables

Scroll down the **System Variables** (bottom window) and doubeclick on the PATH variable

Add the following to the end (make sure you add the initial semicolin)

;C:\PHP

Click OK

**Make PHP.INI available to PHP**

While you still have the **Environment variables** window open click **new**

In the **Variable Name** field enter

PHPRC

Set the **Variable Value** to

C:\PHP

This will make **PHP.INI** available to PHP (We will verify this later)

**Configuring IIS**

You have a choice of whether to setup PHP to use the ISAPI extension, CGI executable, or using FastCGI. The ISAPI extension is not fully stable, and the CGI executable's performance is very poor because after every request the php-cgi.exe executable is unloaded. So if the php-cgi.exe executable is always loaded into memory then that would greatly increase the performance. There are two ways of doing this.  
1. Spend $500 for Zend's own WinEnabler [[2]](http://www.zend.com/store/products/zend-win-enabler.php)  
2. Setup the free FastCGI program that does the same thing as WinEnabler

The recommended way of running PHP on IIS is using FastCGI. Below you will find instructions on how to setup PHP using ISAPI but if your site is going to serve lots of pages, you will probably want to go with FastCGI.

**Add the PHP ISAPI extension to IIS Web Service Extensions**

Click on Start -> Administrative Tools -> Internet Information Services (IIS) Manager

Expand the local computer in the left pane

Click on "Web Service Extensions" in the left pane

In the right pane, click the blue underlined text, "Add a new Web service extension..."

Enter "PHP5 ISAPI" as the "Extension name"

Click the "Add..." button and browse to the php5isapi.dll file in your C:\PHP install directory

Click Open -> OK

Check the "Set extension status to Allowed" checkbox and click "OK"

**Adding the PHP parsing to your IIS website**

***Note:***You can add this either on the top-level **Web Sites** or to individual web sites beneath it. If you add it to the top-level web sites node in the left pane, it applies to **all** websites on the IIS instance. You can also choose to only install it on specific websites beneath the top-level node, in that case it will only apply to that site. The procedure for adding is the same for both scopes.

Be careful when applying this to the top-level node, as it will override settings defined in the individual websites beneath it.

In the left pane, expand **Web Sites**

Right Click the website you want to configure, and select **properties**

Open the **Home Directory** tab

Click **Configuration**

Then go to the **Mappings** tab

Click **Add...**

Enter the full path to **php5isapi.dll** in the "Executable" textbox or click the **Browse** button to browse your way to it. If you have followed the path recommendations in this guide, the fill path should be **C:\PHP\php5isapi.dll**

Enter **.php** in the **Extension** textbox

Select **Limit to**, enter **GET,POST,HEAD**

Click **OK** and verify that **.PHP** is now included in the **Application extensions** listbox

Click **OK**

This configures IIS to understand what to do with files ending with .php

**Adding scripting permissions**

While still having the **Web Site Properties** dialog box open, click **Home Directory**

Make sure that "Execute permissions" dropdown is set to "Scripts only".

Click **OK**

**Setting up FastCGI**

A good set of instructions for setting up FastCGI can be found here [[3]](http://phplens.com/phpeverywhere/fastcgi-php). The basics are below.

Make sure you have PHP 4.3.x or later installed. Earlier versions of PHP require extra work to get FastCGI working.

Download the files at [[4]](http://www.caraveo.com/fastcgi/fastcgi-0.6.zip) and unpack isapi\_fcgi.dll to c:\php\isapi\_fcgi.dll.

Create a file using notepad named fastcgi.reg and insert the following text into it

Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINE\SOFTWARE\FASTCGI]

"StartServers"=dword:00000005

"IncrementServers"=dword:00000002

"MaxServers"=dword:00000019

"Timeout"=dword:00000258

[HKEY\_LOCAL\_MACHINE\SOFTWARE\FASTCGI\.php]

"AppPath"="c:\\php\\php-cgi.exe"

"BindPath"="php-fcgi"

Double click the file you just created to merge those settings into the registry.

Next setup FastCGI the same way as you would the ISAPI filter, except everywhere you see **C:\PHP\php5isapi.dll** you should replace that with **C:\PHP\isapi\_fcgi.dll**

**Testing PHP Installation**

In the left pane, right-click on the website you are working with and select **Properties**

Then go to **Home Directory**

Check the **Local Path**. This shows you were that websites root directory is located (If you are working with the default IIS website, the location will be C:\INETPUB\WWWROOT\)

Create a new file called test.php in that location

The contents of test.php should be:

<?php phpinfo(); ?>

Now, you need to give the anonymous internet user account and NETWORK SERVICE account read/execute permissions on the C:\PHP directory, to allow IIS to parse the .PHP files. You do this by

Perform this for both the IUSR and NETWORK SERVICE accounts

Open **My Computer**

Right-click **C:\PHP**

Select *Properties'*

Go to the **Security** tab

Click *Add*

In the **Enter object names to select** textbox, enter IUSR

Click **Check Names** and it should expand it to NETWORK SERVICE/IUSR\_ServerName

Click *OK'*

Verify that the **Read & Execute**, **List Folder Contents** and **Read** values are set to **Read**.

Click *OK'*

Open the test.php file via your browser, example: *http://www.example.com/test.php*

A page showing your PHP configuration settings should now appear. If not, make sure you have followed **ALL** the steps outlined earlier in the guide.

Verify that PHP uses C:\PHP\PHP.ini by looking at the column for **Configuration File (php.ini) Path** Also, verify that **MySQL** support is loaded, and that the **session\_save.path** is set to the correct values. If not, edit C:\PHP\PHP.ini again, set the values as outlined and then restart the IIS Web Service (IIS Manager->Right Click <servername>(Local computer)->All Tasks->Restart IIS).

After you have verified your PHP install, delete the test.php file from your website.