

106年國民學西班牙語
萬元補助計畫

馬上申請

巨匠線上真

- [Dream.In.Code](#) > [Programming Tutorials](#) > [C++ Tutorials](#)
- (2 Pages)
- ▼
- 1
- 2
- ➡

Introduction to : Creating a Php extension for windows Rate Topic: ★★★★★ 1 Votes

aaron1178

Posted 20 August 2011 - 01:39 AM

Introduction to: Creating an extension for php in windows

Table of Contents

- 1.0 - Introduction
- 1.1 - Php and Zend
- 1.2 - Requirements
- 1.3 - Php 5.3.x Source Code
- 2.0 - Creating a new project
- 2.1 - Additional Directories
- 2.2 - PreProcessor Directives
- 2.3 - Additional Library Directories and Dependencies
- 2.4 - Structure of our extension
- 2.5 - Extension Breakdown
- 2.6 - Complete extension source code
- 3.0 - Building and using our extension
- 4.0 - What's Next

1.0 Introduction

Extensions for Php are like an unknown force, not many people dare venture into the unknown mysteries of extensions for Php. Although people sometimes Write, Design and Deploy extensions, the main contributor for Php extensions are actually the Php developers themselves.

This tutorial is part 1 of an ongoing series published here and when I get around to it, it will also be available on my site [Dark Rising Studios](http://www.darkrisingstudios.com/Development/Tutorials) (<http://www.darkrisingstudios.com/Development/Tutorials>) as a tutorial as per this tutorial and a PDF file.

As mentioned above, this tutorial is an Introduction / Part 1 to the 'Creating an extension for Php in windows' series. This series goes through different features and methods to develop and deploy an extension for Php on the Windows operating system.

On the 3rd August 2010 at 9:19 PM I wrote '[Creating a Php extension in C++](http://www.dreamincode.net/forums/topic/184335-creating-a-php-extension-in-c/)' (<http://www.dreamincode.net/forums/topic/184335-creating-a-php-extension-in-c/>). Since my passion to Write and complete this series grew, I realised that the above tutorial is outdated and ill-explained, so I deem the above tutorial **Out of Date** and inconsistent for learning material.

Ask A Question

Join 500,000 **dream.in.code**® Developers

Follow & Share

Dream.In.Code

G+ Follow



1093 readers
BY FEEDBURNER

C++ Tutorials

- [Microsoft : Shaped Windows](#)
- [Microsoft : Implementing an Indexed Table : Part III](#)
- [Microsoft : Implementing an Indexed Table : Part I and II](#)
- [Processing Data Held In A Comma Separated File](#)
- [The reasons for using vectors](#)

General Discussion ▼

- ▶ Caffeine Lounge
- ▶ Corner Cubicle
- ▶ Student Campus
- ▶ Software Development
- ▶ Industry News
- ▶ Introduce Yourself
- ▶ Nightmare.In.Code

Programming Help ▼

- ▶ C and C++
- ▶ VB.NET
- ▶ Java
- ▶ C#
- ▶ Python
- ▶ PHP
- ▶ Mobile Development
- ▶ ASP.NET
- ▶ .NET Framework
- ▶ Ruby
- ▶ Game Development

Note: Php extensions for the Microsoft OS are stored in Dynamic Linked Libraries or *.dll*

1.1 Php and Zend

Note: If you are a Web Developer and you don't know what Php is, then please crawl out from under that rock you have been living under for millennia and wake up, for crying out loud

Php is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Php is the top-most leading online web development scripting language that is used by more than 69% of websites. More can be found on Php by visiting their website [php.net](http://www.php.net)



(<http://www.php.net>)

Zend has a long history of support for the PHP open-source community. Each year, Zend contributes significant manpower and other resources to help nurture & encourage advancement of both the PHP language and the community. More can be read on that article found here at the Zend website [Zend Engine](http://www.zend.com/en/community/php/)



The PHP Company

(<https://www.zend.com/en/community/php/>)

1.2 Requirements

Note: These requirements are required to successfully complete this series. If you think that you can accomplish this series outside the following requirements, then please so.

In order to successfully follow and complete this series, there is a number of requirements before we can proceed.

- Wamp Server - [Available here for free \(http://www.wampserver.com/en/download.php\)](http://www.wampserver.com/en/download.php) Windows OS
- Php 5.3.5 complete source code [Available here for free \(https://museum.php.net/php5/php-5.3.5.tar.gz\)](https://museum.php.net/php5/php-5.3.5.tar.gz) More on this later
- Microsoft Visual C++ express 2010 [Available here for free \(https://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express\)](https://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express) 2010 edition is needed
- Win Rar [Available here for free \(http://www.rarlab.com/download.htm\)](http://www.rarlab.com/download.htm) Used to uncompress the Php source code

As we do not want to get side tracked from the purpose of this series, I will show you the **How-to** for the Php source code and explain why we need it.

1.3 Php 5.3.x Source Code

Once you have installed Wamp Server, Visual Studio C++ express 2010 and Win Rar you will need to extract the Php source code to "Wamp Install Directory/bin" where "Wamp Install Directory" is where Wamp Server installed to Generally "C:/Wamp". For reference, please view Fig1 and Fig2.

- [Introduction to C++ Metaprogramming: Basics](#)
- [Pointers, and a possible problem - if you're not careful!](#)
- [Generating Random Numbers - The C++ Way](#)
- [Hello World: Your first C and C++ Programs](#)
- [Network programming under UNIX](#)
- [350 More C++ Tutorials...](#)

Reference Sheets



Code Snippets

- [C Snippets](#)
- [C++ Snippets](#)
- [Java Snippets](#)
- [Visual Basic Snippets](#)
- [C# Snippets](#)
- [VB.NET Snippets](#)
- [ASP.NET Snippets](#)

► Assembly
► Databases
► ColdFusion
► VB6
► Other Languages
► 52 Weeks Of Code


Web Development
► Web Development
► HTML & CSS
► JavaScript
► Graphic Design
► Flash & ActionScript
► Blogging
► SEO & Advertising
► Web Servers & Hosting
► Site Check

Posted Image
Fig1

Posted Image
Fig2

We need the source code of our current wamp version of php. This is how we compile our extension so that it uses the correct up-to-date methods/functions.

You will also need a copy of config.w32.h file that for some reason does not come with Php 5.3.x or above, I downloaded the Php 5.2.17 and extracted it via Win Rar to "C:\Wamp\bin\php-5.3.5\main". You can find the Php-5.2.17 available in Php's archives [here for free download \(http://au.php.net/distributions/php-5.2.17.tar.gz\)](http://au.php.net/distributions/php-5.2.17.tar.gz). Since you only need the one file, I have included a Win Rar file containing the config.w32.h file.

 (http://www.dreamincode.net/forums/index.php?app=core&module=attach§ion=attach&attach_id=25274&s=36edfba59c43a246c1a60aeca5ea2aac) config.w32.h.zip
(http://www.dreamincode.net/forums/index.php?app=core&module=attach§ion=attach&attach_id=25274&s=36edfba59c43a246c1a60aeca5ea2aac) (2.12K)
Number of downloads: 2371

2.0 Creating a new project

We want to create a new project so we can begin on writing, developing and deploying our extension so it is available for Php. To do this, open up Visual Studio C++ express 2010 and create a new project. This is accomplished via "File -> New Project" or "New Project" on the start page.

Select Win32 Project and enter a name. For this chapter I am using "Introduction", please refer to Fig 3.

Posted Image

Fig 3

Click the "next" button and it will bring you to a new form with the title of "Win32 Application Wizard - Introduction". Name of project. Now make sure under the heading "Application Type" that the **DLL** is selected and **Empty Project** under "Additional Options" is also selected, please refer to Fig 4.

Posted Image

Fig 4

We now have our project created and ready to configure to our needs.

Before we can continue on, we need to add a source file. To do this, either click on Source Files under your project and go Right Click -> Add -> New Item. Select C++ file (.cpp) and enter a name. For this extension I am adding the name `return_string` with a suffix of `php_` so in total, "**php_return_string**". This is so the properties of the file are set to default and is able to be edited via properties. Refer to 2.1. Refer to Fig 5

Posted Image

Fig 5

2.1 Additional Directories

We need to add some "Additional Directories" so that our IDE Visual Studio C++ can include and use methods / functions inside our Php extension. To do this we go to properties. To navigate to Properties,

- [PHP Snippets](#)
- [Python Snippets](#)
- [Ruby Snippets](#)
- [ColdFusion Snippets](#)
- [SQL Snippets](#)
- [Assembly Snippets](#)
- [Functional Programming Snippets](#)
- [Perl Snippets](#)
- [HTML/CSS Snippets](#)
- [Javascript Snippets](#)
- [Flash/ActionScript Snippets](#)
- [ASP Snippets](#)
- [Linux, Unix, and Bash Snippets](#)
- [Other Languages Snippets](#)
- [Regex](#)

DIC Chatroom

■ [Join our IRC Chat](#)

Bye Bye Ads

DIC++

either Right click introduction Or what ever name you named the project -> Properties , or Project -> Properties, refer to Fig 6

Posted Image

Fig 6

Now navigate to **C/C++ -> General -> Additional Include Directories**. Now select the arrow to the right It's a dropdown box -> **Edit Items**. Now a screen titled **Additional Include Directories** will come up. To the far top right, there is a buttons panel, to the button to the left ^{The file button}, Click it, now add in the following directories, one by one ^{Click the file icon for every directory given}

```
1 C:\Wamp\bin\php-5.3.5\
2 C:\Wamp\bin\php-5.3.5\main
3 C:\Wamp\bin\php-5.3.5\TSRM
4 C:\Wamp\bin\php-5.3.5\win32
5 C:\Wamp\bin\php-5.3.5\Zend
```

It does not matter in which order you implement into the Additional Include Directories field, Refer to Fig 7

Posted Image

Fig 7

Okay, we have added the Additional Directories so that we may use methods / functions from Php and Zend.

2.2 PreProcessor Directives

PreProcessor Directives are just global variable for compilation time. The following directives will tell our build environment ^{And Php and Zend} that we want to configure our DLL with Php and Zend Windows 32-bit configuration. We also tell that the Zend debugger is set to off.

```
1 PHP_WIN32
2 ZEND_WIN32
3 ZTS=1
4 ZEND_DEBUG=0
```

To add the directives, please navigate to properties ^{Refer 2.0}, and click on **C/C++ -> PreProcessor -> PreProcessor Directives** and add ^{Refer to 2.0} the above under **Introduction_exports**. Refer to Fig 8

Posted Image

Fig 8

These are all the PreProcessor Directives that we need for this chapter. So we now have:



- New project named "**Introduction**"
- 5 Additional Include Directories
- 4 PreProcessor Directives

So in 2.3, we will add Additional Libraries and Dependencies to our Introduction extension.

2.3 Additional Library Directories and Dependencies

When creating an extension, be sure to remember to complete this step. This will reference our extension to the current Php version and functions that have already been built. This is to allow of extension to communicate with Php. Not following this step will mass errors.

Navigate to **Properties** -> **Linker** -> **General** -> **Additional Library Directories** and add the following to it. Please reference Fig 9 to do so.

C:\Wamp\bin\php\php5.3.5\dev

Posted Image

Fig 9

The above directory has to be the current Php version that our extension uses with Wamp.

Now, to the dependencies. This will add the file that we need. It finds the file in the above directory and creates a dependency to it.

php5ts.lib Note that earlier versions of Php use php5.lib

Navigate to **Properties** -> **Linker** -> **Input** -> **Additional Dependencies** and add the following to it. Please reference Fig 10 to do so.

Posted Image

Fig 10

2.4 Structure of our extension

In this section of the tutorial, the full source code for our extension will be posted here and you will be given a complete line-by-line description on what happens and what it does.

Please note, that after trying a few different methods to return a string parameter, no luck was found. So our extension will return a pre-defined string depending on a boolean value (true / false)

```

15
16     STANDARD_MODULE_HEADER,
17     "Use Html",
18     use_functions,
19     NULL, NULL, NULL, NULL, NULL,
20     "1.0.0-tutorial",
21     STANDARD_MODULE_PROPERTIES
22 };
23
24 ZEND_GET_MODULE(use_html);
25

```

```
26 ZEND_FUNCTION(use_html)
27 {
28     bool useHtml;
29
30     if(zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "|b", &useHtml) == FAILURE)
31     {
32         E_ERROR;
33         return;
34     }
35
36     if(useHtml)
37     {
38         php_printf("This string uses <a href='#'>Html</a>");
39     }
40     else
41     {
42         php_printf("This string does not Html");
43     }
44
45     return;
46 }
```

2.5 Extension Breakdown

Now that you have had a glance at the extension code, it doesn't really require a lot of code to write simple extensions. In this chapter, we will dissect our extension line-by-line. Lets get to it shall we.

```
1 #define PHP_COMPILER_ID "VC6"
```

This line tells our extension to compile with Php with the defined PreProcessor directive (PHP_COMPILER_ID). When build time comes, this allows our extension to be 'silently' built with Visual C++ 6. What I mean by silent is, that it tricks Php in thinking that we are building our extension with Visual C++ 6, which we are not, we are building with a later version of Visual C++.

```
1 #include "php.h"
```

Well, it is as it seems. This line tells the compiler to scan all directories to look for the Php header file. Which can be found in **C:\wamp\bin\php-5.3.5\main**

```
1 ZEND_FUNCTION(long_return);
```

`ZEND_FUNCTION()`; declares a new C function that complies with Zend's internal API. This takes 1 known parameter which as of date, is the functions name, which will be named **long_return**

```
1 zend_function_entry use_functions[] =
```

Now that we have declared the functions to be exported, you also have to introduce them to Zend. Introducing the list of functions is done by using an array of `zend_function_entry`'s. This array contains all functions that are to be made available externally, with the function's name as it should appear in Php.

```
1 ZEND_FE(use_html, NULL)
2 {NULL, NULL, NULL}
```

`ZEND_FE(function_name, arg_types)` defines a function entry of the name `function_name` in `zend_function_entry`. It requires a corresponding C function We declared it in the last line of code above. `arg_types` has to be set to `NULL`.

You can see that the last entry in the list always has to be `{NULL, NULL, NULL}`. This marker has to be set for Zend to know when the end of the list of exported functions is reached.

```
1 zend_module_entry use_html_module_entry =
```

This defines a zend module for our extension. We **HAVE** use our function name followed by **module_entry** or this will throw errors at build. It contains all necessary information to describe the contents of this module to Zend.

```
1 STANDARD_MODULE_HEADER,
2 "Use Html",
3 use_functions,
4 NULL, NULL, NULL, NULL, NULL,
5 "1.0.0-tutorial",
6 STANDARD_MODULE_PROPERTIES
```

STANDARD_MODULE_HEADER contains four bits of important information, instead of filling out the four bits, we use `STANDARD_MODULE_HEADER`. This will fill out the four bits of information that we need More on this in another chapter so we do not need to do so.

"Use Html" is the name of the module. It is a string value that will also show up in the phpinfo() table More on this later. This value **is** required.

use_functions is array of functions we defined beforehand. We use the exact name of the defined structure in order for zend to do its magic. This is a must have parameter.

NULL, NULL, NULL, NULL, NULL These are parameters that we do not need, so we set them as unused, NULL. More on this in a later chapter.

1.0.0-tutorial This is the version of the module / function **NOT** the extension. You can use **NO_VERSION_YET** if you don't want to give a version yet, but it is really recommend that you add a version string here.

STANDARD_MODULE_PROPERTIES This is all parameters of the remaining module. We use **STANDARD_MODULE_PROPERTIES** because it is recommended to do so, unless we use startup / shutdown functions, which we do not.

```
1 ZEND_GET_MODULE(use_html);
```

ZEND_GET_MODULE()'s purpose is to pass the module information back to Zend in order to inform the engine about the module contents. This is a required line, or zend will throw yet another error.

```
1 ZEND_FUNCTION(use_html)
```

Implementing the exported functions is the final step

The function declaration is done using **ZEND_FUNCTION**. After the declaration, code for checking and retrieving the function's arguments, argument conversion, and return value follows.

```
1 bool useHtml;
```

If you do not understand this line, it's ok, all this will do is create a boolean ^{true / false} variable named **useHtml**. This is so we can store the parameter value of our function inside our **useHtml** variable.

```
1 if(zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "|b", &useHtml) == FAILURE)
```

I'm not going to go into detail with the if statement, you should already know what it is and how to use it.

zend_parse_paramerters() The first argument to this function is supposed to be the number of parameters passed to your function, so **ZEND_NUM_ARGS()** can be used for that. The second parameter should always be the **TSRMLS_CC** macro.

The third argument is a string that specifies the number and types of arguments your function is expecting. We use **|b**, where **|** = optional and **b** = boolean value. So **|b** means an optional boolean

parameter.

The final argument(s) are pointers to variable to store parameter data in, in this case, we store a boolean value in our **useHtml** variable.

FAILURE This calls the FAILURE macro that is determines the success of failure of the if statement. So if we do not receive anything buy optional boolean, it will return FAILURE or true in other senses.

```
1 E_ERROR;
2 return;
```

E_ERROR This will throw an E_ERROR message to Php if the enclosing **if** statement is true.

return If the enclosing if statement is true, it will return it from our function so we cannot display any more data. This is useful if we want to print a string and have it exit our function.

```
1 php_printf("This string uses <a href='#'>Html</a>");
```

if the enclosing if statement is true If our useHtml variable is true, then this will call a php function that will print a pre-defined string.

2.6 Complete extension source code

Additional Directories

```
1 C:\Wamp\bin\php-5.3.5\
2 C:\Wamp\bin\php-5.3.5\main
3 C:\Wamp\bin\php-5.3.5\Zend
4 C:\Wamp\bin\php-5.3.5\Win32
5 C:\Wamp\bin\php-5.3.5\TSRM
```

PreProcessor Directives

```
1 PHP_WIN32
2 ZEND_WIN32
3 ZTS=1
4 ZEND_DEBUG=0
```

Additional Library Directories

```
1 C:\Wamp\bin\php\php5.3.5\dev\
```

Additional Dependencies

```
1 php5ts.lib
2 OR
3 php5.lib
```

Extension Source Code

```
01 #define PHP_COMPILER_ID "VC6"
02
03 #include "php.h"
04
05
06 ZEND_FUNCTION(use_html);
07
08 zend_function_entry use_functions[] =
09 {
10     ZEND_FE(use_html, NULL)
11     {NULL, NULL, NULL}
12 };
13
14 zend_module_entry use_html_module_entry =
15 {
16     STANDARD_MODULE_HEADER,
17     "Use Html",
18     use_functions,
19     NULL, NULL, NULL, NULL, NULL,
20     "1.0.0-tutorial",
21     STANDARD_MODULE_PROPERTIES
22 };
23
24 ZEND_GET_MODULE(use_html);
25
26 ZEND_FUNCTION(use_html)
27 {
28     bool useHtml;
29
30     if(zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "|b", &useHtml) == FAILURE)
31     {
```

3.0 Building and using our extension

Now that we have our extension written, it is now time for us to build it and use it in Php. Lets begin the final process of this chapter by building our project. This is accomplished via **Debug -> Build Solution** or the F7 key. Your project is now built, if not, start this tutorial again. Refer to Fig 11 for more information.

Posted Image

Fig 10

Now that our extension has been built into a Dynamic Extension (.dll), open the wamp tray manager and stop **ALL** services. Refer to Fig 11

Posted Image

Fig 11

Once all Wamp services have stopped, navigate to the debug folder of your extension. Find where it says Introduction.dll ^{The name of our extension}. If you cannot find it, the console in Visual Studio C++ express will have the direct path to our extension.

Once you have found our extension, copy it and navigate to **C:\Wamp\bin\php\php5.3.5\ext** where this is the directory of Wamps current running Php version. Paste the extension into the file containing the rest of the extensions.

Now we need to edit php.ini. To do so, we need to go to **Wamp Manager -> PHP -> php.ini** and click it. This should bring it up in notepad.exe for default. Scroll down just over half way to the Dynamic Extensions section, of hit **Ctrl F** and enter extension= and add the following line to the end of the list.

```
1 extension=Introduction.dll
```

Where Introduction.dll is the name of our project built in Visual Studio. Now start ALL of Wamp's services. This should start strait away. If you get errors, review the Apache log.

Now that our extension is loaded in Php, it's time to check that Php is actually using our extension. If not, redo the whole tutorial.

Create a Php page inside Wamp's WWW directory and add the following to it.

```
1 <?php
2 phpinfo();
3 ?>
```

Load the page in a web browser <http://localhost/test.php> (<http://localhost/test.php>) for me and scour the entire page looking for our extension name ^{Use Html}. If it is there, Php is using it :) Refer to Fig 12.

Posted Image

Fig 12

Because we named it **Use Html** it will generally be more towards the bottom as all extensions are sorted in

alphabetical order.

Now on the same Php page, replace the text with

```
1 <?php
2
3 use_html(true);
4
5 ?>
```

We should now see "This string uses Html". If we set the parameter to false or do not parse a parameter will see "This string does not Html"

It's alive

4.0 What's Next

Well, now that's over, you are probably wondering what's next right? Well for those hoping for more tutorials on this rare topic, there will be more forthcoming tutorials. Once the series is finished, you will be able to write your own extensions that could be the next big thing for Php :) Well hopefully anyway.

I have had a lot of fun writing this tutorial and hope you join me for the series,

Thanks,

Aaron Thompson

Replies To: Introduction to : Creating a Php extension for windows

Atli

Posted 07 November 2011 - 04:45 PM

Hey.

Great tutorial! :)

For people using VC9 builds of Apache and PHP, you will get errors when compiling the extension code in the tutorial. To fix it, try these two steps:

- Instead of copying the config.w32.h file from a 5.2 version of the code, or using the one [aaron1178](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=121914) (<http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=121914>) provides, copy the win32\build\config.w32.h.in file from the 5.3 source and save it as main\config.w32.h.
- Change the PHP_COMPILER_ID definition in the extension source code to read "VC9" instead of "VC6".

aaron1178

Posted 07 November 2011 - 07:19 PM

Sorry about these errors guys. I wasn't aware of them as I must be using an older version of their source code. Atli has provided an adequact way of solving the issues.

Thanks for the feedback Atli

codeprada

Posted 14 November 2011 - 04:10 PM

Nice tutorial. Definitely going to try this out.

yunter

Posted 25 December 2011 - 08:06 PM

oh, Great tutorial!

Why image file can't see

abxccd

Posted 10 April 2012 - 04:35 AM

Is there anyway to get the extension to compile as non-threadsafe?

olibenu

Posted 10 April 2012 - 06:23 AM

nice!

it's even a tutorial on how to present a tutorial!

aaron1178

Posted 11 April 2012 - 01:36 AM

☺ (<http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1596673>) abxccd, on 10 April 2012 - 10:35 PM, said:
Is there anyway to get the extension to compile as non-threadsafe?

To be honest abxccd (<http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=3535>), I am not 100% entirely sure. I am leaning towards no.

☺ (<http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1596726>) olibenu, on 11 April 2012 - 12:23 AM, said:
nice!
it's even a tutorial on how to present a tutorial!

Why thank you :)

bahman2216

Posted 09 December 2012 - 06:50 AM

thanks for good article

I just have an error during compile via php 5.2.9 but it is work good on php 5.3.x

here is my error in php 5.2.9
c:\program files\microsoft visual studio 9.0\vc\include\sys\stat.inl(44) : error C2466: cannot allocate an array of constant size 0

nobelb

Posted 13 December 2012 - 06:53 PM

Thanks, Great tutorial!

If some one get error likes "error LNK2019: unresolved external symbol"? may mean that you are compiling a x64 library in a Win32 project.

How to compile a 64-bit application using Visual C++ 2010 Express
<http://stackoverflow.com/questions/1865069/how-to-compile-a-64-bit-c-2010-express>

[application-using-visual-c-2010-express\)](#)

I help this can help some people.

karimkhan_it

Posted 19 June 2013 - 08:32 AM

My project build successfully and I copied dll fine in appropriate folder.

Even it is giving function not defined error and in case of phpinfo(); nothing about use_html!


Atli

Posted 20 June 2013 - 11:38 PM

[karimkhan_it \(http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=457330\)](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=457330), we need more details than that. It's pretty much impossible to know what's wrong without knowing exactly what's going on.


aaron1178

Posted 04 July 2013 - 06:57 PM

 [\(http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1864383\)](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1864383) karimkhan_it, on 20 June 2013 - 02:32 AM, said:

My project build successfully and I copied dll fine in appropriate folder.

Even it is giving function not defined error and in case of phpinfo(); nothing about use_html!

 [\(http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1864384\)](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&app=forums&module=forums§ion=findpost&pid=1864384) karimkhan_it, on 20 June 2013 - 02:32 AM, said:

Fatal error: Call to undefined function tutorial() in D:\wamp\www\test.php on line

My project build successfully and I copied dll fine in appropriate folder.

Even it is giving function not defined error and in case of phpinfo(); nothing about tutorial!

[karimkhan \(http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=201325\)](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=201325), notice that the line in bold uses a function tutorial, please do not use any material from:
<http://www.dreamincode.net/forums/topic/184335-creating-a-php-extension-in-c>

It's old and a bad tutorial I wrote.

As [Atli \(http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=391059\)](http://www.dreamincode.net/forums/index.php?s=05f9ee23685e43d10501d2139efed472&showuser=391059) explained, we need more details. Are you in windows? or linux?. Are you enabling the extension from the .ini? We need to know what steps you took, after you compiled the .dll file... and we need to know where you placed it.

This post has been edited by **aaron1178**: 04 July 2013 - 06:58 PM

alebonfa

Posted 15 September 2013 - 04:43 AM

Hi, Aaron,

Thanks for your article. It's a fantastic and detailed article.
I've tested it and It's worked perfectly.

I have one only doubt:

I've downloaded the php_dbase source code in php.net.
How can I join this source code in this project?

I need to compile que php_dbase.dll to last php version (5.5.3).

Thanks,
Alexandre Bonfa

StephenIsienyi

Posted 12 October 2013 - 06:02 PM

Hi Aaron,

This is an awesome article.

There are a couple questions though - hopefully they are minor.

1) What should I do to get around a requirement issue? My PHP version is 5.3.10 instead of the 5.3.5 you used. I also do not have the main, TSRM, Win32 or Zend directories in my C:\Wamp\bin\php\php-5.3.10 folder

2) What other modifications would I need to make to successfully port this extension from Windows to a Linux successfully?

Thanks,
• (2 Pages)

- ▼
- 1
- 2
- →

Related C++ Topics^{beta}

- [Creating A Php Extension In C++ - Ever Wanted To Know How? Tutorial](#)
- [Program To Implement Insertion Sort Algorithm For The Given Inputfile.Program To Implement Insertion Sort Algorithm For The Given Inputfile.](#)
- [Looking For Windows SDK August 2001 - Looking For Windows SDK August 2001 Download Or Place Of Purchase](#)
- [\[link\] VS Express 2012 For Windows Desktop Is Here!](#)
- [Building Php Extension Using C - How To Build Php Extension Using C For Class And Object In Php](#)
- [How To Put Column On A For Loop - Having Trouble Creating A Column Within A For Loop](#)
- ["Hello, World" For Windows Shell Extension?](#)
- [EXE File Data Extension - Writing To And Reading From Data Extension In EXE File](#)

- [Driver Development For Windows 64-bit - Questions Concerned With The Release Of Drivers For 64-bit Versions Of](#) **Tutorial**
- [Why Microsoft Has Made Developers Horrified About Coding For Windows 8](#)

[FAQ](#) | [Team Blog](#) | [Feedback/Support](#) | [Advertising](#)  | [Terms of Use](#) | [Privacy Policy](#) | [About Us](#)

Copyright 2001-2017 [MediaGroup1 LLC](#), All Rights Reserved
A [MediaGroup1 LLC](#) Production - Version 6.0.2.1.36
Server: secure3