



Uploading Images With PHP Build The Next Flickr! (DICr???)

Rate Topic: ★★★★★ 7 Votes

akozlik

#1



D.I.C Addict



Reputation: 90

Posts: 797
Joined: 25-February 08**Intermediate** Posted 10 December 2008 - 12:41 PM

There have been a lot of posts in the PHP forum regarding how to upload files to a server and put them into a database. A lot of people were trying to put the entire binary file into the database, which was the incorrect way to do things. This tutorial is an attempt to explain how to properly upload images to your website, and store their records in the database using PHP and MySQL.

Please note that this tutorial has **a lot** of code. The code is well commented and will explain much of what we're trying to do. I'll give an overview in the tutorial, but you need to dig into the code to really understand how things work. Also, please note that this tutorial assumes you know some basics of PHP and MySQL and don't have any problems writing and reading to and from a database.

The first thing we'll do is build a database. I have created a simple database named 'people', which will hold first and last names of people, as well as a picture of them. You can copy and paste this SQL statement into phpMyAdmin to build your table.

```
1 CREATE TABLE `people` (  
2   `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
3   `fname` VARCHAR( 30 ) NOT NULL ,  
4   `lname` VARCHAR( 40 ) NOT NULL ,  
5   `filename` VARCHAR( 50 ) NOT NULL  
6 ) ENGINE = MYISAM ;
```

Next, we will build our directory structure. We have our root folder which will hold all our directories, as well as our PHP files. Within this root directory we have a directory named 'images' and another named 'includes'. When we finish we will have a structure that looks like this

```
1 root-directory/images.php  
2 root-directory/index.php  
3 root-directory/upload.php  
4 root-directory/images/  
5 root-directory/includes/conn.php
```

Ask A Question

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

Lead the path to IT innovation.
Windows Server 2003 support
is ending July 14, 2015.
[Upgrade now](#)

Follow & Share

Dream.In.Code



BY FEEDBURNER

General Discussion ▾

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

PHP Tutorials

- [Basic Form Validation](#)
- [Securing login forms from brute-force attacks using queues.](#)
- [Display Formatted Code](#)
- [Simple Content Formatting with Regular Expressions](#)
- [Limiting login attempts](#)
- [CodeIgniter Text Based Mafia Game || Login/Usersonline part 3](#)
- [Adding Your Own Functionality To Smarty w/ Plugins](#)
- [CodeIgniter Text Based Mafia Game || Register part 2](#)
- [CodeIgniter Text Based Mafia Game || Intro / Set-up Part 1](#)

Programming Help ▾

- C and C++
- VB.NET
- Java
- C #
- ASP.NET
- .NET Framework
- VB6
- PHP
- Ruby
- Python
- ColdFusion
- Databases
- Other Languages
- Game Development
- Mobile Development
- 52 Weeks Of Code

Web Development ▾

- Web Development
- HTML & CSS
- JavaScript

Within the includes/conn.php file, you should paste this code:

```
01 <?php
02
03 // Input your information for the database
   here
04
05 // Host name
06 $host = "localhost";
07
08 // Database username
09 $username = "username";
10
11 // Database password
12 $password = "password";
13
14 // Name of database
15 $database = "db_name";
16
17 $conn = mysql_connect($host, $username,
   $password) or die ("Could not connect");
18 $db = mysql_select_db($database, $conn) or die
   ("Could not select DB");
19
20 ?>
```

On any page that connects to the database, you will require() this file. Be sure to save that file as conn.php within the includes folder.

Next we build our index file. For us, this will just be a form with three fields. One will be 'fname', one will be 'lname', and the next will be a file upload named 'image'. Please note I did some basic CSS styling, but if you don't want to style it that's fine.

```
01 <?php
02 // Start a session for displaying any form
   errors
03 session_start();
04 ?>
05 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
   Strict//EN"
06 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
   strict.dtd">
07
08 <html xmlns="http://www.w3.org/1999/xhtml">
09   <head>
10     <title>Dream in code
       tutorial</title>
11
12     <style type="text/css">
13       label
14       {
15         float: left;
16         text-align: right;
17         margin-right: 10px;
18         width: 100px;
19         color: black;
20       }
21
22
```

- [Functions that return booleans for fun and profit.](#)
- [170 More PHP Tutorials...](#)

Reference Sheets



Code Snippets

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

DIC Chatroom

- [Join our IRC Chat](#)

Bye Bye Ads

DIC++

Find us on Facebook



67,923 people like Dream.In.Code.



Facebook social plugin

- ▶ Graphic Design
- ▶ Flash & ActionScript
- ▶ Blogging
- ▶ SEO & Advertising
- ▶ Web Servers & Hosting
- ▶ Site Check

Make sure you notice that we are starting a session at the very top of the page. This will be handy for reporting any errors we have with the form. In addition, we have some code before the form to display any errors that are encountered. After displaying the error message we unset the error session variable.

Another thing to note is the `MAX_FILE_SIZE`. This can be used to ensure that the filesize of the file uploaded is less than that variable. I've left you on your own to write that portion of the script, but I promise you it is not hard.

Save that file as `index.html` in your root directory.

Now comes the important part. Here is the entire source file with comments. I will briefly go over how it works, but the details are within the source code:

```
01 <?php
02 // Start a session for error reporting
03 session_start();
04
05 // Call our connection file
06 require("includes/conn.php");
07
08 // Check to see if the type of file uploaded
09 // is a valid image type
09 function is_valid_type($file)
10 {
11     // This is an array that holds all the
12     // valid image MIME types
12     $valid_types = array("image/jpg",
13                          "image/jpeg", "image/bmp", "image/gif");
14
14     if (in_array($file['type'],
15                 $valid_types))
16         return 1;
16     return 0;
17 }
18
19 // Just a short function that prints out the
20 // contents of an array in a manner that's easy
20 // to read
20 // I used this function during debugging but
```

Alright, so we start out by starting a session and requiring our `conn.php` file. Nothing new there.

Next, we have a function that checks to see if the upload file is of a valid type. This function helps deter people from uploading exe files, as well as other potentially harmful stuff you don't want on your server. We have a variable that holds all the valid MIME types. That list is by no means exhaustive, but it works for our purposes. For a full list of MIME types check out the [Webmaster Toolkit's list](#)

The next function is just something I use for debugging. This prints the contents of an array so that it's readable.

We create a constant variable that will hold our target path. I didn't use the `define()` function, but I could have.

Next block of code checks to make sure none of the fields were blank. This is just very basic form validation and can be adjusted for what you see fit.

The next block makes a call to the `is_valid_type` function and checks to make sure that the file is valid.

The next block of code checks to see if a file of that name already exists, in an effort to keep people from overwriting files. A good way to ensure a unique filename is to parse out the name part of the file name before the period, append a timestamp, and then hash it. After hashing it, add the extension back on. I'm attempting to keep this tutorial simple, so I did not show that, but you should be able to figure that out. Just use substrings.

Finally, we attempt to move the uploaded file to the new directory. Make sure the new directory has proper write permissions, or your script won't be able to save.

If moving the file was successful, we input the information into the database. Make sure you notice that we **do not input the binary file into the database**. What we have done is moved the file to a directory where we can access it, and then stored the images file name into the database. Whenever we want to print out the image we get its file name from the database and then build the `img` tag.

Lastly, we have our `images.php` page, which will display all those great images you've uploaded.

```
01 <?php
02 // Get our database connector
03 require("includes/conn.php");
04 ?>
05 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
06 Strict//EN"
07 "http://www.w3.org/TR/xhtml1/DTD/xhtml1:
08 strict.dtd">
09
10 <html xmlns="http://www.w3.org/1999/xhtml">
11 <head>
12 <title>Dream in code tutorial - List
13 of Images</title>
14 </head>
15 <body>
16
17 <div>
18
19 <?php
20 // Grab the data from our
21 people table
22 $sql = "select * from
23 people";
24 $result = mysql_query($sql)
```

There's really nothing there that you shouldn't be able to understand.

Well, that's all I have for you! Now you know how to

properly upload images to a website and store their **references** into the database. You get two thumbs up ☐ ☐.

If you've found this tutorial useful, or if you have any comments let me know! Good luck!



Is This A Good Question/Topic? **12**

Replies To: Uploading Images With PHP

Moonbat

#2



D.I.C Regular

Author

D.I.C++

Reputation: 36



Posts: 424

Joined: 30-June 08

Posted 16 December 2008 - 07:12 PM

I'm glad you brought up the whole "references" thing. I can't beleive how many people have thought of trying to store the image itself in the DB. This'll be a great help for newbies (hey, it was for me! ☐)

Was This Post Helpful? **0**

Robsphp

#3



New D.I.C Head

Reputation: 0



Posts: 3

Joined: 10-January 09

Posted 10 January 2009 - 05:13 PM

akozlik

THANK YOU, I am trying to add to my members database an option to have there own avi file then grab it out of the database and display it.

Would I use the same idea (upload the avi's into a directory then reference them in the users table) ?

regards

Rob

Was This Post Helpful? **0**

akozlik

#4



D.I.C Addict

Posted 10 January 2009 - 06:56 PM

Robsphp, on 10 Jan, 2009 - 07:13 PM, said:

akozlik



Reputation: 90



Posts: 797
Joined: 25-February 08

THANK YOU, I am trying to add to my members database an option to have there own avi file then grab it out of the database and display it.

Would I use the same idea (upload the avi's into a directory then reference them in the users table) ?

regards

Rob

Yeah the idea's the same. I'm not sure how to best handle partially streamed videos or anything like that, but you can get the party started by using the same idea.

Hope that helps.

Was This Post Helpful? 0

Quote + Reply MultiQuote

RobspHP

#5



New D.I.C Head

Reputation: 0



Posts: 3
Joined: 10-January 09

Posted 11 January 2009 - 02:07 PM

Akozlik

I get the following error ?

Warning: Cannot modify header information - headers already sent by (output started at /public_html/test/includes/conn.php:10) in /public_html/test/upload.php on line 81

When i try the images.php it seems to have worked but I get that error when upload.php is used ????

Rob

Was This Post Helpful? 0

Quote + Reply MultiQuote

teampoop

#6



D.I.C Head

Contributor

DIC++

Reputation: 17



Posts: 140
Joined: 19-June 08

Posted 11 January 2009 - 10:49 PM

You have to have your header() calls before you do any output to the browser. So if your code has an echo (or print) before your header call, you will get an error. This is probably why there was a session variable set, to pass the information to the redirected page.

ex:

```
1 <?php
2 echo "checking for stuff";
3 header("Location: index.php"); // ERROR ensues
4 ?>
```

Hope this helps.

RobspHP, on 11 Jan, 2009 - 01:07 PM, said:

Akozlik

I get the following error ?


Warning: Cannot modify header information - headers already sent by (output started at /public_html/test/includes/conn.php:10) in /public_html/test/upload.php on line 81

When i try the images.php it seems to have worked but I get that error when upload.php is used ????

Rob

Was This Post Helpful?  

0

 Quote + Reply

 MultiQuote

RobspHP  



New D.I.C Head

Reputation: 0



Posts: 3

Joined: 10-January 09

Posted 12 January 2009 - 03:01 AM

#7


Still not working ?

So does the session variable have to be cleared ?

Is the code wrong ?

regards

Rob


 **teampoop, on 11 Jan, 2009 - 09:49 PM, said:**

You have to have your header() calls before you do any output to the browser. So if your code has an echo (or print) before your header call, you will get an error. This is probably why there was a session variable set, to pass the information to the redirected page.

ex:

```
1 <?php
2 echo "checking for stuff";
3 header("Location: index.php"); // ERROR ensues
4 ?>
```

Hope this helps.

 **RobspHP, on 11 Jan, 2009 - 01:07 PM, said:**

Akozlik

I get the following error ?

Warning: Cannot modify header information - headers already sent by (output started at /public_html/test/includes/conn.php:10) in /public_html/test/upload.php on line 81

When i try the images.php it seems to have worked but I get that error when upload.php is used ????

Rob

Was This Post Helpful?  

0

 Quote + Reply


 MultiQuote

akozlik 

#8



D.I.C Addict

 Expert

 DIC++

Reputation: 90



Posts: 797

Joined: 25-February 08

Posted 12 January 2009 - 12:16 PM

The session variable does not need to be cleared.

Are you printing **anything** before you redirect? That includes any html tags, spaces, whatever. If you send anything to the browser, your header won't be sent.

If you start a thread in the PHP Forum and post your code, we can take a look. I'd like to keep comments on here specifically focused on the tutorials themselves. Each individual case can be handled in the thread.

Thanks and glad you like the tut!

Was This Post Helpful?  

0

 Quote + Reply

 MultiQuote

goodmuyis 

#9



New D.I.C Head

Reputation: 0



Posts: 9

Joined: 13-September 08


Posted 24 February 2009 - 02:55 AM

Hello Mr, thanks for this stuff is just for me but am trying to change the file size but it doesn't respond to my changes, the size is in byte or Kb(am sorry for that)

and thanks for ur stuff once again

Was This Post Helpful?  

0

 Quote + Reply

 MultiQuote

chingkit 

#10



New D.I.C Head

Reputation: 0



Posts: 1

Joined: 09-May 09

Posted 09 May 2009 - 09:23 AM

Hi,reading this tutorial really caught my attention. I am now working on my personal website, and planning to add my own photo gallery..I try to experiment with the code you have given and I find it so cool and easy to understood to a newbie like me. I Just like ask for one part of the tutorial, because I'm a bit confused. How can I create the directory.. Is it the folder which I'll be using to store all of my php files?..something like this.

```
root-directory/images.php
root-directory/index.php
root-directory/upload.php
root-directory/images/
root-directory/includes/conn.php
```

Hoping for your reply.Thanks in Advance=)

This post has been edited by **chingkit**: 09 May 2009 - 09:50 AM

Was This Post Helpful?  

0

[Quote + Reply](#)[MultiQuote](#)**Jupiter** 

#11



New D.I.C Head

Reputation: 0



Posts: 30

Joined: 01-February 07

Posted 08 August 2009 - 10:55 AM

Thanks a lot for this nice tutorial..it works like a charm Was This Post Helpful?  

0

[Quote + Reply](#)[MultiQuote](#)**rkmalbz** 

#12



New D.I.C Head

Reputation: 0



Posts: 38

Joined: 10-August 09

Posted 17 August 2009 - 04:08 PM

Next, we will build our directory structure. We have our root folder which will hold all our directories, as well as our PHP files. Within this root directory we have a directory named 'images' and another named 'includes'. When we finish we will have a structure that looks like this

```
1 root-directory/images.php
2 root-directory/index.php
3 root-directory/upload.php
4 root-directory/images/
5 root-directory/includes/conn.php
```

I'm wondering whats the use of that root-directory..... and how do i put it up?? or how to build the directory?? sorry need some quick basic explanation.

This post has been edited by **rkmalbz**: 17 August 2009 - 04:13 PM

Was This Post Helpful?  

0

[Quote + Reply](#)[MultiQuote](#)**neos300** 

#13



D.I.C Head

Reputation: 2



Posts: 62

Joined: 21-April 09

Posted 20 August 2009 - 05:57 AM

The root directory is the directory that you will put all your .php files in.

Was This Post Helpful?  

0

[Quote + Reply](#)[MultiQuote](#)**kewlcreator** 

#14



D.I.C Lover

Author

★ DIC++

Reputation: 38

Posted 25 September 2009 - 12:23 PM

Thanks so much! It was fun to make!

Root directory:

You really don't have to worry about this. This is the folder where you will put all your files. Since the name of the directory is not needed in this script (other than to view the



Posts: 1,065
Joined: 25-March 09

finished files), this is just another word for "any folder".

This post has been edited by **kewlcreator**: 25 September 2009 - 12:25 PM

Was This Post Helpful?

0

Quote + Reply

MultiQuote

mulson

#15



New D.I.C Head

Reputation: 1



Posts: 11
Joined: 21-September 09

Posted 05 October 2009 - 04:56 PM

Robsphp, on 10 Jan, 2009 - 04:13 PM, said:

akozlik

THANK YOU, I am trying to add to my members database an option to have there own avi file then grab it out of the database and display it.

Would I use the same idea (upload the avi's into a directory then reference them in the users table) ?

regards

Rob

i like the tutorial,but it is possible to display only one image at a time,not neccessary all...this is because i will like to dispaly only a profile picture for each user,when they login



akozlik, on 10 Dec, 2008 - 11:41 AM, said:

There have been a lot of posts in the PHP forum regarding how to upload files to a server and put them into a database. A lot of people were trying to put the entire binary file into the database, which was the incorrect way to do things. This tutorial is an attempt to explain how to properly upload images to your website, and store their records in the database using PHP and MySQL.

Please note that this tutorial has **a lot** of code. The code is well commented and will explain much of what we're trying to do. I'll give an overview in the tutorial, but you need to dig into the code to really understand how things work. Also, please note that this tutorial assumes you know some basics of PHP and MySQL and don't have any problems writing and reading to and from a database.

The first thing we'll do is build a database. I have created a simple database named 'people', which will hold first and last names of people, as well as a picture of them. You can copy and paste this SQL statement into phpMyAdmin to build your table.

```
1 CREATE TABLE `people` (  
2   `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
3   `fname` VARCHAR( 30 ) NOT NULL ,  
4   `lname` VARCHAR( 40 ) NOT NULL ,  
5   `filename` VARCHAR( 50 ) NOT NULL  
6 ) ENGINE = MYISAM ;
```

Next, we will build our directory structure. We have our root folder which will hold all our directories, as well as our PHP files. Within this root directory we have a directory named 'images' and another

named 'includes'. When we finish we will have a structure that looks like this

```
1 root-directory/images.php
2 root-directory/index.php
3 root-directory/upload.php
4 root-directory/images/
5 root-directory/includes/conn.php
```

Within the includes/conn.php file, you should paste this code:

```
01 <?php
02
03 // Input your information for the database here
04
05 // Host name
06 $host = "localhost";
07
08 // Database username
09 $username = "username";
10
11 // Database password
12 $password = "password";
13
14 // Name of database
15 $database = "db_name";
16
17 $conn = mysql_connect($host, $username, $password)
18 or die ("Could not connect");
19 $db = mysql_select_db($database, $conn) or die
20 ("Could not select DB");
21
22 ?>
```

On any page that connects to the database, you will require() this file. Be sure to save that file as conn.php within the includes folder.

Next we build our index file. For us, this will just be a form with three fields. One will be 'fname', one will be 'lname', and the next will be a file upload named 'image'. Please note I did some basic CSS styling, but if you don't want to style it that's fine.

```
01 <?php
02 // Start a session for displaying any form
03 session_start();
04 errors
05 ?>
06 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
07 Strict//EN"
08 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
09 strict.dtd">
10
11 <html xmlns="http://www.w3.org/1999/xhtml">
12 <head>
13 <title>Dream in code tutorial</title>
14
15 <style type="text/css">
16 label
17 {
18 float: left;
19 text-align: right;
20 margin-right: 10px;
21 width: 100px;
22 color: black;
23 }
24
25 #submit
26 {
27 float: left;
28 margin-top: 5px;
29 position: relative;
30 left: 110px;
31 }
32
33 #error
34 {
```

Make sure you notice that we are starting a session at the very top of the page. This will be handy for reporting any errors we have with the form. In addition, we have some code before the form to display any errors that are encountered. After displaying the error message we unset the error session variable.

Another thing to note is the `MAX_FILE_SIZE`. This can be used to ensure that the filesize of the file uploaded is less than that variable. I've left you on your own to write that portion of the script, but I promise you it is not hard.

Save that file as `index.html` in your root directory.

Now comes the important part. Here is the entire source file with comments. I will briefly go over how it works, but the details are within the source code:

```
01 <?php
02 // Start a session for error reporting
03 session_start();
04
05 // Call our connection file
06 require("includes/conn.php");
07
08 // Check to see if the type of file uploaded is
09 // a valid image type
10 function is_valid_type($file)
11 {
12     // This is an array that holds all the valid
13     // image MIME types
14     $valid_types = array("image/jpg",
15         "image/jpeg", "image/bmp", "image/gif");
16
17     if (in_array($file['type'], $valid_types))
18         return 1;
19     return 0;
20 }
21
22 // Just a short function that prints out the
23 // contents of an array in a manner that's easy to
24 // read
25 // I used this function during debugging but it
26 // serves no purpose at run time for this example
27 function showContents($array)
28 {
29     echo "<pre>";
30     print_r($array);
31     echo "</pre>";
32 }
33
34 // Set some constants
```

Alright, so we start out by starting a session and requiring our `conn.php` file. Nothing new there.

Next, we have a function that checks to see if the upload file is of a valid type. This function helps deter people from uploading exe files, as well as other potentially harmful stuff you don't want on your server. We have a variable that holds all the valid MIME types. That list is by no means exhaustive, but it works for our purposes. For a full list of MIME types check out the [Webmaster Toolkit's list](#)

The next function is just something I use for debugging. This prints the contents of an array so that it's readable.

We create a constant variable that will hold our target path. I didn't use the `define()` function, but I could have.

Next block of code checks to make sure none of the fields were blank. This is just very basic form validation and can be adjusted for what you see fit.

The next block makes a call to the `is_valid_type` function and checks to make sure that the file is valid.

The next block of code checks to see if a file of that name already exists, in an effort to keep people from overwriting files. A good way to ensure a unique filename is to parse out the name part of the file name before the period, append a timestamp, and then hash it. After hashing it, add the extension back on. I'm attempting to keep this tutorial simple, so I did not show that, but you should be able to figure that out. Just use substrings.

Finally, we attempt to move the uploaded file to the new directory. Make sure the new directory has proper write permissions, or your script won't be able to save.

If moving the file was successful, we input the information into the database. Make sure you notice that we **do not input the binary file into the database**. What we have done is moved the file to a directory where we can access it, and then stored the images file name into the database. Whenever we want to print out the image we get its file name from the database and then build the img tag.

Lastly, we have our `images.php` page, which will display all those great images you've uploaded.

```
01 <?php
02 // Get our database connector
03 require("includes/conn.php");
04 ?>
05 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
06 Strict//EN"
07 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
08 strict.dtd">
09
10 <html xmlns="http://www.w3.org/1999/xhtml">
11 <head>
12 <title>Dream in code tutorial - List of
13 Images</title>
14 </head>
15 <body>
16 <div>
17 <?php
18 // Grab the data from our people
19 table
20 $sql = "select * from people";
21 $result = mysql_query($sql) or
22 die ("Could not access DB: " . mysql_error());
23
24 while ($row =
25 mysql_fetch_assoc($result))
26 {
27     echo "<div
28     class=\"picture\">";
29     echo "<p>";
30
31 // Make that we can build the
```

There's really nothing there that you shouldn't be able to understand.

Well, that's all I have for you! Now you know how to properly upload images to a website and store their **references** into the database.


You get two thumbs up ☐ ☐.

If you've found this tutorial useful, or if you have any comments let me know! Good luck!

i like the tutorial, but it is possible to display only one image at a time, not necessary all... this is because i will like to display only a profile picture for each user, when they login.

Was This Post Helpful?  

1

 Quote + Reply

 MultiQuote

[Previous Topic](#) | [PHP Tutorials](#) | [Next Topic](#)

(3 Pages)  1 2 3 

Cloud Database System

trackvia.com/Cloud-Database

Create Your Own Cloud Database. Easy, Powerful & 100% Free to Try!



Related PHP Topics^{beta}

- [Creating A File Upload Form With PHP](#)
- [Help On Uploading Images With Php Please](#)
- [Uploading Images With GIF Extension](#)
- [Image Uploading In PHP](#)
- [Fckeditor With Php - How To Use Fck Editor With Php](#)
- [Tpl Files With Php - How To Map Tpl Files With Php](#)
- [Jquery-charts In Combination With PHP - How To Draw An Easy Jquerychart In Combination With PHP?](#)
- [How To Connect Crystal Reports With PHP? - Hi, I Wanted To Know How We Can Connect Crystal Reports With Php. Actua](#)
- [Store Image Or Url In The Database+ Php](#)
- [Replace Pdf With Another Pdf](#)

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

Copyright 2001-2014 [MediaGroup1 LLC](#), All Rights Reserved
A [MediaGroup1 LLC](#) Production - Version 6.0.2.1.36
Server: secure3