
Subject: Code for downloads
Posted by [Distrbd21](#) on Tue, 09 Dec 2008 04:19:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Is there a code for downloads like i see on some sites.

Downloaded x ammount of times?

I have been googlen but still can't find anything.

Subject: Re: Code for downloads
Posted by [Genesis2001](#) on Tue, 09 Dec 2008 04:23:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Uhm. When someone downloads a file off a file database, the php script would usually update a column in your MySQL database that contains a count of downloads a certain file on your database has received...

Subject: Re: Code for downloads
Posted by [IronWarrior](#) on Tue, 09 Dec 2008 04:25:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes.

Game-Maps.NET uses a not so great system to log "downloads" wish it was better, but meh.

Subject: Re: Code for downloads
Posted by [Genesis2001](#) on Tue, 09 Dec 2008 04:54:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

IronWarrior wrote on Mon, 08 December 2008 21:25Yes.

Game-Maps.NET uses a not so great system to log "downloads" wish it was better, but meh.

*cough*paFailDB*cough*

Subject: Re: Code for downloads
Posted by [witebolx](#) on Thu, 11 Dec 2008 10:05:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

What do we need
Apache compiled with mod_rewrite (off by default, you need to add --enable-module=rewrite to

your configure line)
 PHP as the scripting engine
 A database, like MySQL, to store the download data
 Apache's configuration
 Options +FollowSymLinks
 RewriteEngine On
 RewriteBase /foobar/
 RewriteRule download/send.php - [L]
 RewriteRule download/(.+)\$ download/send.php?file=\$1 [L]
 You can put this block of code in a .htaccess file in /foobar/, your downloads should be in /foobar/download/ - these are the directories accessible by the paths mentioned from your webserver.

What we do is switch on FollowSymLinks which is required by mod_rewrite, if you don't need other options you can remove the +. We turn on the rewrite engine which is off by default, then set the base location for the URI rewrites.

The next two lines are our rewrite rules, if the request is for send.php (our script that counts downloads) we don't modify it, if we get a request for download/foo.bar that will become download/send.php?file=foo.bar for Apache (the rewrite base is prepended). The rule processed only filenames with extensions.

The download counter

```
<?php
```

```
$file = isset($_GET['file']) ? trim($_GET['file']) : '';
```

```
if (!$file) {
    die("Error");
}
```

```
if ( substr_count($file, '..') > 0 or substr($file, 0, 1) == '/' ) {
    die("Invalid filename.");
}
```

```
$path = dirname($_SERVER['PATH_TRANSLATED']) . '/' . $file;
```

```
if ( !file_exists($path) ) {
    die("File not found: $file");
}
```

```
$ext = explode('.', $file);
if ( sizeof($ext) < 2 ) {
    die("Invalid filename: should have extension");
}
```

We do some checking first, you should never display files to the visitors that they have requested without checking for unwanted characters like ../ or / at the start of the filename.

The \$path's value is set to the directory containing our script + the filename requested. We then check if the file really exists and if it has an extension.

```
$ext = $ext[sizeof($ext)-1];
```

```
switch ($ext) {  
  case 'tgz' :  
    $type = 'application/x-gzip';  
    break;  
  
  case 'php' :  
    $type = 'text/html';  
    break;  
  
  default :  
    $type = 'text/plain';  
}
```

```
header("Content-type: $type");
```

By default PHP sends a content type header of text/html to the browser so we need to modify it when this is not right. You should add more extension -> MIME type pairs if you serve different file types. We've sent text/html for PHP files because we want to present them syntax highlighted - our script is something like a download counter + PHP file browser.

Because we send a HTTP header there should be nothing sent to the browser before the last line of the block above. Output buffering can be used as a way around this but it's not needed here.

```
$fd = fopen ($path, "r");  
$code = fread ($fd, filesize($path));  
fclose ($fd);
```

```
switch ($ext) {  
  case 'php' :  
    ?>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
  "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">  
<html>  
  <head>  
    <title><?php echo $file?> syntax highlighted</title>  
  </head>  
  <body>  
<?php  
  highlight_string($code);  
?>  
  </body>  
</html>  
<?php
```

```
break;

default :
    echo $code;
}
```

This is the part which sends the file to the browser or presents the highlighted PHP file. The functions used are binary safe so you can send every type of files not only text.

```
require_once('.././config.php');
$db = db_connect();
```

```
$sql = "UPDATE download_file SET count=count+1 WHERE file = '$file' ";
```

```
$db->query($sql);
?>
```

And the final one, which actually counts the download. Include a file with our database info first - as you can see if opens a file which is out of the webserver root which makes it pretty safe. We connect to the database with our predefined function `db_connect()` which returns a `PEAR::DB` instance.

The query updated the `download_file` table, which holds the download files and the times they have been downloaded. We increase the count field by one for our file.

Note: Never make the mistake to first `SELECT` the count value, increment it in PHP and then write it to the database.