.htaccess

- How to use suPHP on my webspace
- How to redirect my site from HTTP to HTTPS?
- Change HTTP header settings using .htaccess file
- URL redirect/rewrite using the .htaccess file
- What is the .htaccess file and what do I use it for?
- How to Change your Default Index Page in htaccess
- How to change my document root folder using an .htaccess file?
- How to create .htaccess file?

How to use suPHP on my webspace

WHAT IS THE SUPHP?

suPHP is a tool for executing PHP scripts with the permissions of their owners. It consists of an Apache module (mod_suphp) and a setuid root binary (suphp) that is called by the Apache module to change the uid of the process executing the PHP interpreter.

Works with the PHP versions available in the PHP version option on the cPanel interface.

If you don’t want to use the default PHP version under on your domain name, you need to create a .htaccess file in a particular domain name assigned to a folder (document_root) will be placed into one of the following lines. After setting recursively, all sub-directories will be applied to the setting.

PHP version 4.4:

```
AddHandler application/x-httpd-suphp44 .php .php5 .php4 .php3
```

PHP version 5.1:

```
AddHandler application/x-httpd-suphp51 .php .php5
```

PHP version 5.2:

```
AddHandler application/x-httpd-suphp52 .php .php5
```

PHP version 5.3:

```
AddHandler application/x-httpd-suphp53 .php .php5
```

PHP version 5.4:

```
AddHandler application/x-httpd-suphp54 .php .php5
```

PHP version 5.5:

```
AddHandler application/x-httpd-suphp55 .php .php5
```

PHP version 5.6:

```
AddHandler application/x-httpd-suphp56 .php .php5
```

PHP version 7.0:

```
AddHandler application/x-httpd-suphp70 .php .php5
```

PHP version 7.1:

```
AddHandler application/x-httpd-suphp71 .php .php5
```

PHP version 7.2:

```
AddHandler application/x-httpd-suphp72 .php .php5
```

PHP version 7.3:

```
AddHandler application/x-httpd-suphp73 .php .php5
```

PHP version 7.4:

```
AddHandler application/x-httpd-suphp74 .php .php5
```

PHP version 8.0:

```
AddHandler application/x-httpd-suphp80 .php .php5
```

PHP version 8.1:

```
AddHandler application/x-httpd-suphp81 .php .php5
```

The module enabled per PHP version is a global setting, but you can change your php.ini settings with a .php.ini file placement. Starting with PHP 5.4 version, you can put a .user.ini file, each of which can override PHP settings for that folder (and its subdirectories).

How to redirect my site from HTTP to HTTPS?

REDIRECT ALL WEB TRAFFIC

To force all web traffic to use HTTPS insert the following lines of code in the .htaccess file in your website’s root folder.

```
RewriteCond %{REQUEST_URI} !^/[0-9]+\..+\.cpaneldcv$
RewriteCond %{REQUEST_URI} !^/\.well-known/pki-validation/[A-F0-9]{32}\.txt(?:\ Comodo\ DCV)?$
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.example.com/$1 [R=301,L]
```

Be sure to replace *www.example.com*with your actual domain name.

REDIRECT ONLY SPECIFIED DOMAIN

To force a specific domain to use HTTPS, use the following lines of code in the .htaccess file in your website’s root folder:

```
RewriteCond %{REQUEST_URI} !^/[0-9]+\..+\.cpaneldcv$
RewriteCond %{REQUEST_URI} !^/\.well-known/pki-validation/[A-F0-9]{32}\.txt(?:\ Comodo\ DCV)?$
RewriteEngine On
RewriteCond %{HTTP_HOST} ^example\.com [NC]
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://www.example.com/$1 [R=301,L]
```

Make sure to replace *example\.com* with the domain name you’re trying force to https. Additionally, you need to replace *www.example.com* with your actual domain name.

REDIRECT SPECIFIED FOLDER

If you want to force SSL on a specific folder you can insert the code below into a .htaccess file placed in that specific folder:

```
RewriteCond %{REQUEST_URI} !^/[0-9]+\..+\.cpaneldcv$
RewriteCond %{REQUEST_URI} !^/\.well-known/pki-validation/[A-F0-9]{32}\.txt(?:\ Comodo\ DCV)?$
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteCond %{REQUEST_URI} folder
RewriteRule ^(.*)$ https://www.example.com/folder/$1 [R=301,L]
```

Make sure you change the *folder* reference to the actual folder name. Then be sure to replace *www.example.com/folder* with your actual domain name and folder you want to force the SSL on.

Change HTTP header settings using .htaccess file

First step, if you do not already have a .htaccess file created in the root directory of the affected domain name document. To do this, refer to the following:

How to create .htaccess file?

If the file already exists, you can edit it as described below:

- Creating and editing a file via FTP
- Creating and editing a file via SSH

WHAT IS THE HTTP HEADER?

The HTTP header is part of an HTTP request or response. This determines the operating parameters of the HTTP transaction. For more information, see the following link:

List of HTTP header fields

The .htaccess file can be used to modify or complement the HTTP response header.

CHARSET HEADER

The .htaccess file can use the following to force the header of a given content type. The charset header specifies the document's character encoding. You can add the header without the meta tag:

```
<IfModule mod_headers.c>
   AddDefaultCharset UTF-8
   AddDefaultCharset ISO-8859-2
</IfModule>
```

CONTENT-LANGUAGE HEADER

In the .htaccess file, you can set a language header as follows. You can add the header without the meta tag:

```
<IfModule mod_headers.c>
   DefaultLanguage hu-hu
</IfModule>
```

CACHE-CONTROL HEADER

Cache-Control is one of the most common headers used for websites. This determines how long the file is stored in your browser

For example, if you set 5 minutes in the Cache-Control header, the visitor’s browser downloads the page and then caches it for 5 minutes. After 5 minutes, the page must be retrieved from the server.

For example:

In the following example, we set the web page to be stored for 5 minutes by visitors browsers.

```
<IfModule mod_headers.c>
   Header set Cache-Control "max-age=300, public"
</IfModule>
```

Syntax

max-age is set in seconds.

The caching policy may be “public”, “private” or “no-store”.

USE ‘VARY’ HTTP HEADERS FOR MOBILE PAGES

The following Google article describes the use of Vary headers for mobile pages:

- The Vary HTTP Header

SEURITY

CONTENT-SECURITY-POLICY

Content-Security-Policy header helps reduce XSS risks. For more details, see the following pages:

- https://content-security-policy.com/

STRICT-TRANSPORT-SECURITY (HSTS)

Specifies that browsers will only communicate over HTTP instead of HTTPS. For more details, see the following pages:

- https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
- https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet

Enable the following content in the .htaccess file:

```
<IfModule mod_headers.c>
   Header add Strict-Transport-Security "max-age=31415926;includeSubDomains;"
</IfModule>
```

You can testing the following command:

```
curl -I https://example.com
```

The output looks like this:

```
[server]$ curl -I https://example.com
HTTP/1.1 200 OK
Date: Tue, 05 Jun 2018 20:05:52 GMT
Server: Apache
Last-Modified: Tue, 05 Jun 2018 16:26:52 GMT
ETag: "2f9-56de78493cbc8"
Accept-Ranges: bytes
Content-Length: 761
Strict-Transport-Security: max-age=31415926;includeSubDomains;
Content-Type: text/html
```

The command output shows the Strict-Transport-Security header

URL redirect/rewrite using the .htaccess file

By default your website can be accessed with both www.example.com and example.com. Since Google penalizes this due to duplicated content reasons, you should restrict the access to either www.example.com or example.com. Some links may be outside of your website scope and/or the search engines may have already indexed your website under both addresses.

- Using the Redirect tool in the cPanel

REDIRECTING TO OR FROM WWW

How do I redirect all links for www.example.com to example.com?

Create a 301 redirect forcing all http requests to use either www.example.com or example.com:

- **Redirect example.com to www.example.com:**

```
RewriteEngine On
RewriteCond %{HTTP_HOST} !^www.example.com$ [NC]
RewriteRule ^(.*)$ http://www.example.com/$1 [L,R=301]
```

- **Redirect www.example.com to example.com:**

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^www\.example\.com$
RewriteRule ^/?$ "http\:\/\/example\.com\/" [R=301,L]
```

EXPLANATION

The first line tells apache to start the rewrite module.
The next line:

```
RewriteCond %{HTTP_HOST} !^www.example.com$ [NC]
```

specifies that the next rule only fires when the http host (that means the domain of the queried url) is not (specified with the `!`) www.example.com.

The `$` means that the host ends with www.example.com and the result is that all pages from www.example.com will trigger the following rewrite rule. Combined with the inversive `!` is the result

every host that is not www.example.com will be redirected to this domain.

The `[NC]` specifies that the http host is case insensitive. The escapes the `.` because this is a special character (normally, the dot (.) means that one character is unspecified).

The final line describes the action that should be executed:

```
RewriteRule ^(.*)$ http://www.example.com/$1 [L,R=301]
```

The `^(.*)$` is a little magic trick. Can you remember the meaning of the dot? If not, this can be any character(but only one).

So `.*` means that you can have a lot of characters, not only one.

This is what we need because `^(.*)$` contains the requested url, without the domain.

The next part `http://www.example.com/$1` describes the target of the rewrite rule. This is our “final” used domain name, where $1 contains the content of the `(.*)`.

The next part is also important, since it does the 301 redirect for us automatically: `[L,R=301]`. `L` means this is the last rule in this run. After this rewrite the webserver will return a result. The `R=301` means that the webserver returns a 301 moved permanently to the requesting browser or search engine.

HOW TO REDIRECT VISITORS TO HTTPS?

In case you set a valid certificate for a domain name, we recommend that visitors redirect an unsafe http:// address to secure https://.

You can easily redirect the redirection by modifying the .htaccess file in your document_root folder for that domain name.

Use the following rows in this example to redirect all users of a domain name from insecure (http://) URLs to secure (https://) URLs.

Replace the example.com domain name with your domain name.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule (.*) https://example.com/$1 [R=301,L]
```

What is the .htaccess file and what do I use it for?

Htaccess files are hidden plain text files that are on the server to help control how your visitors interact with your website. The htaccess file is also used to block specific traffic from being able to view your website. If you look for your .htaccess file you'll see that there's no filename. The extension is .htaccess which tells the server what type of file it is. In cPanel you can see if you have a current .htaccess file using file manager but you will need to make sure you have selected to view hidden files. To view hidden files in file manager, select the **File manager** icon in cPanel and make sure the box is checked next to **Show Hidden Files.** Then click **OK** and you will be able to view hidden files.

WHAT CAN YOU DO WITH A .HTACCESS FILE?

You might have a private area of your website you wish to keep password protected. This password protection is actually set up in the .htaccess file. Most of the functions of the htaccess file, you do not have to concern yourself with as they will be automatically written through cPanel. This is the case of password protecting directories. While you set it up in cPanel, it actually writes a directive to your htaccess file.

Other functions of the htaccess file include, prohibiting hotlinks, rewriting URLs, setting default pages, creating redirects, reconfiguring account settings, and much more. It's really important to realize how the htaccess file can affect your entire account. Changing something in the htaccess file can alter how your website functions so it's really important BEFORE making changes to your htaccess to backup your current htaccess file.

TROUBLESHOOTING ERRORS CAUSED BY THE .HTACCESS FILE

If you are getting errors on your website, the .htaccess file can often be the culprit.

This is easily tested by renaming your current htaccess file. Often, during troubleshooting I'll simply rename the .htaccess to .htaccess.old and now I'll reload the website. If the site loads I then know the issue resides in my configuration of the .htaccess file. If it does not fix the issue I was having, I'll rename the htaccess by removing the .old I added to the end. That way, it won't affect my website after I resolve the issue.

How to Change your Default Index Page in htaccess

The server looks for specifically named files as the first page of your website, also known as the index page. The default order of index file names our particular servers look through is index.php, index.htm, index.html, and finally default.htm. You can change the name of the index file your account looks for by altering the .htaccess file. Perhaps you want to have a specific custom name for your index file or maybe you are migrating from another host and the index page is named differently. This way your internal links will not be broken by renaming the index file.

What is the .htaccess file and what do I use it for?

CHANGING YOUR DEFAULT INDEX FILE VIA HTACCESS

If you have not created a .htaccess file in the document_root directory for the domain name then you have to type it. File creation or modification can be done with cPanel's built-in file manager or any FTP client or SSH protocol. You can find a more detailed description of this through the following links:

- Creating and editing a file via cPanel internal File Manager
- Create and modify a file via FTP
- Creating and editing a file via SSH

Paste the following code at the top of the page to configure your desired index page. In our example below, we decided to make the index page of our folders named first.html.

```
#Alternate default index page
DirectoryIndex first.html
```

You can also list more than one file in the configuration. The file will be read left to right and check for them in that order. In this example, we add index.htm, index.html, and index.php to the list. First the server will check for first.html, if it does not find a file with that name, it continues to index.htm and so on.

```
#Alternate default index pages
DirectoryIndex first.html index.htm index.html index.php
```

How to change my document root folder using an .htaccess file?

By default your website is loaded from the *public_html* folder of your account.
The *public_html* directory is also called web root folder or document root folder.
If you've created a test website under a sub-folder and you want it to be displayed when you type your domain name, add the following lines to the *.htaccess* file in the *public_html* folder:

```
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domain-name.com$ [NC,OR]
RewriteCond %{HTTP_HOST} ^www.domain-name.com$
RewriteCond %{REQUEST_URI} !folder/
RewriteRule (.*) /folder/$1 [L]
```

In the above lines you should replace the following 2 strings:

domain-name.com – Type your own domain name
folder – Type the name of the sub-folder which has the test/development website

How to create .htaccess file?

You can create an .htaccess directly on your web server using an FTP client, cPanel internal File Manager or SSH. View the following articles for instructions on how to use either option to create the file:

- Creating and editing a file via FTP (easier for beginners)
- Creating and editing a file via SSH (for advanced users)

If you're using an FTP client, make sure it has been configured to show hidden files. This is necessary since the .htaccess file begins with a period.

Make sure that when you create the .htaccess file you do NOT add a file extension. This file should only be titled .htaccess with no extension.

WHAT PERMISSIONS SHOULD THE FILE HAVE?

644 permissions are usually fine for an .htaccess file. When you create the file on the server, it should already have these permissions set, so there is most likely nothing to change.

WHERE TO PUT YOUR .HTACCESS FILE?

Generally, you put the .htaccess file in your website's main directory. However, the location ultimately depends on what you're attempting to do with the .htaccess file.

The .htaccess file can control behavior for every directory under the folder it resides in. For this reason it can also be in your user's home directory. Again, it depends on what you're using the .htaccess file for.