# Customizing WordPress Via the ".htaccess" File

WordPress (WP) has become a popular Content Management System (CMS) for creating a wide variety of websites. The interface provides newbies as well as veterans with a user-friendly dashboard for designing and customizing any website. An infinite variety of add-on applications, called plugins, are also available to add functionality. WP also supplies abundant choices, both free and paid, for template designs called "themes." These themes offer easy integration of a host of website formats including:

- Informational
- eCommerce
- Blogs
- Portfolio

The WP interface offers a myriad of customizable options. In addition to themes, plugins, and dashboard options, there is another way to customize and add functionality to your WP website. Every WP download will provide a Hypertext Access (.htaccess) file that creates a window into the backend of your online creation.

The .htaccess file is a configuration file. It is used by Apache-based web servers, where most Content Management Systems (CMS) are hosted. The most popular CMS interfaces include WordPress, Drupal, and Joomla. When using any of these CMS interfaces you will encounter the .htaccess file.

Whether you've come across the name .htaccess in previous publications, or the term is completely new to you, fret not. This article will cover all you need to know, from the basics to more advanced options.

The .htaccess file is a simple, directory-level, text file. The information on the file controls tasks in the folder or "directory" where it is located, along with any subdirectories located in the same folder as the file. When updating the .htaccess file, remember the "dot" at the beginning of the file. Unlike many other web files, the name of the file does not have an extension. Again, the dot is found at the beginning of the name and must remain there for the file to function properly.

Learning to manipulate the .htaccess file of your website provides additional control options. Via the .htaccess file you can alter the behavior of your website or affect one specific directory of your website. If the file is stored in the root directory, changes will affect the entire site. By placing a copy with additional instructions into a specific directory location, for instance the content folder, will cause an alterations to only affect that one directory.

Through the .htaccess file you can:

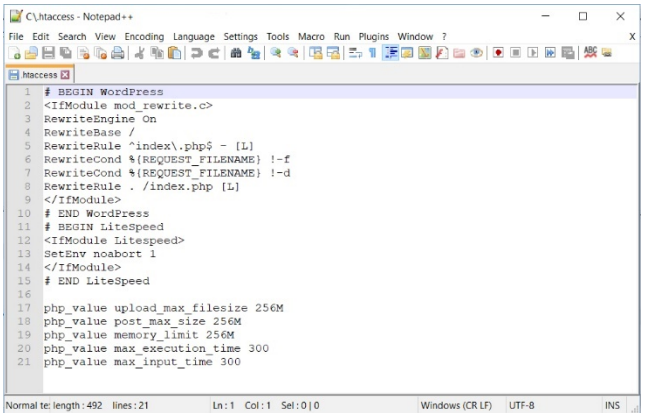
- Protect Access via Passwords
- Block access to your website based on an incoming IP address
- Alter which page is used as your landing page (other than index.html)
- Prevent directory listing when an index.html file does not exist
- Redirect visitors to other pages
- Customize Error Pages
- Enable Server Side Includes (SSI) directives to add information like the current time
- Add Multi-purpose Internet Mail Extensions (MIMEs)

This important file is used to configure access issues including URL redirection and shortening, control of site access, and more. It will pay off in the long run to learn a little bit of backend WP customization. The .htaccess file is the perfect place to start.

The .htaccess file and several others are the core of your WP website's operations. By default, this file holds information relating to how the website's permalinks are displayed. When you update your permalink preferences through the WP dashboard, the file is automatically updated with the changes. Unless additional alterations are made, this may be the file's only function.

The .htaccess file can be used to create 301 redirects which alters the web address (URL) of a page as it is appears in search results. With this important file you can also increase your site's security by restricting access to important files and folders and the site itself. When plugins are activated on the dashboard many add code to the .htaccess to include their functionality on the website.

Let's get familiar with the .htaccess file. Below you'll find an image of the file's contents. Some files will appear slightly different.




*# BEGIN WordPress*
*<IfModule mod_rewrite.c>*
*RewriteEngine On*
*RewriteBase /*
*RewriteRule ^index\.php$ - [L]*
*RewriteCond %{REQUEST_FILENAME} !-f*
*RewriteCond %{REQUEST_FILENAME} !-d*
*RewriteRule . /index.php [L]*
*</IfModule>*
*# END WordPress*
*# BEGIN LiteSpeed*
*<IfModule Litespeed>*
*SetEnv noabort 1*
*</IfModule>*
*# END LiteSpeed*

*php_value upload_max_filesize 256M*
*php_value post_max_size 256M*
*php_value memory_limit 256M*
*php_value max_execution_time 300*
*php_value max_input_time 300*

This may seem like gibberish but don't worry. We will explain this file to you, that's what we're here for. Adding tasks and functions through the file does not require an IT degree or even understanding of all the techie details. We will lead you step by step and help you create a duplicate of the original file, in case there is ever an issue with the alterations. Just follow these instructions to find the file, add provided code, and save the changes.

We are required to tell you that making changes to core files on a website can be risky. Adding or removing code from any core file can disrupt the normal operation of the site or interrupt its "live" status on the internet completely. Follow the steps listed exactly. Copy and paste snippets when possible and you'll have no issues.

There are a few things that are required (highly recommended) before you begin altering core files. To avoid any problems:

- Back up your site
  - We recommend the free version of the "Updraft Plus" user-friendly plugin
- Download a copy of the .htaccess file to your computer
  - Duplicate the file and never alter the original (instructions follow)
  - Never edit the "Original" file so that you can always start over
- Test your edits
  - Check your site for changes after each edit
  - If a problem occurs go back to the last edit and correct or delete the code
  - If you are able, a staging site (not live) is the perfect way to test edits

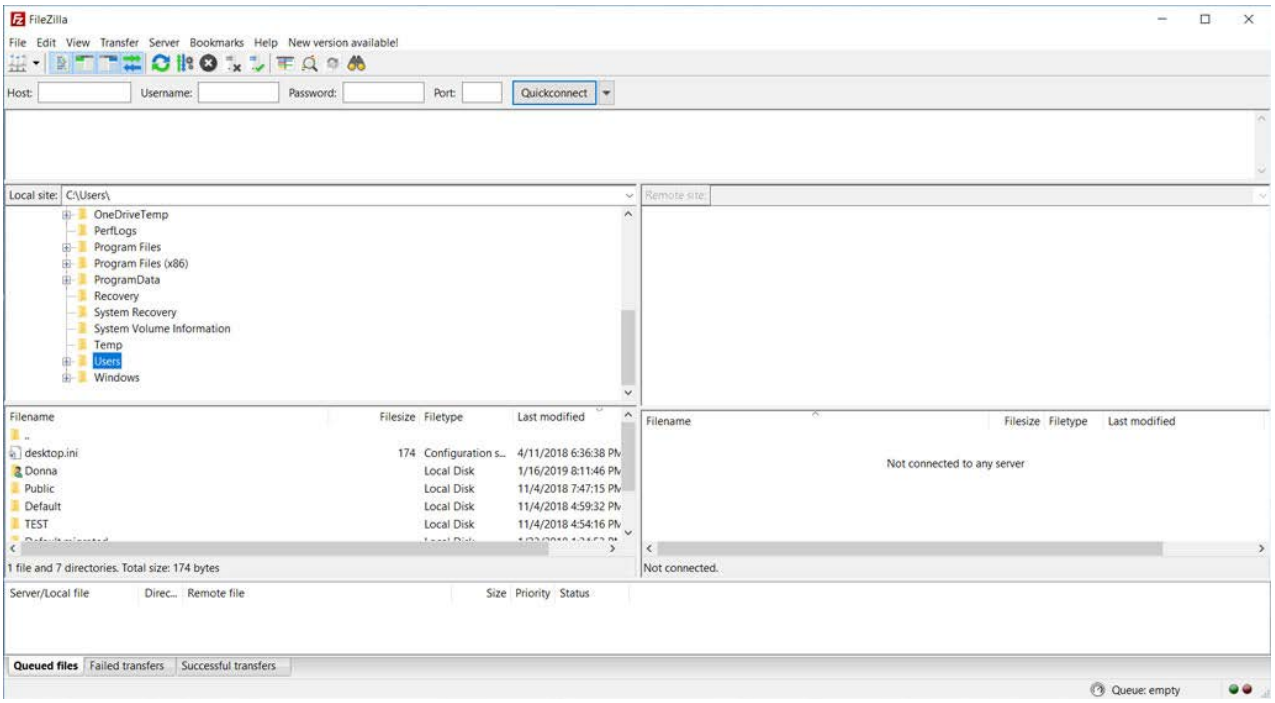
## Step 1: Locate the .htaccess file

To locate the file, you will need to access your website through a File Transfer Protocol (FTP) program.
An FTP program connects your computer to your website's files on your host server. Linux hosting services are compatible with .htaccess files. While Unix systems recognize the file, files that begin with a "dot" are hidden. To locate the file you will have to enable "view hidden files."

In some cases, your web hosting package will not allow the .htaccess file to be edited directly, and we don't recommend it. That's one major reason we recommend using an FTP program to access the file. Through FTP you can:

- create a duplicate file
- edit the copy offline
- upload to your desired location
- test to be sure changes have taken affect

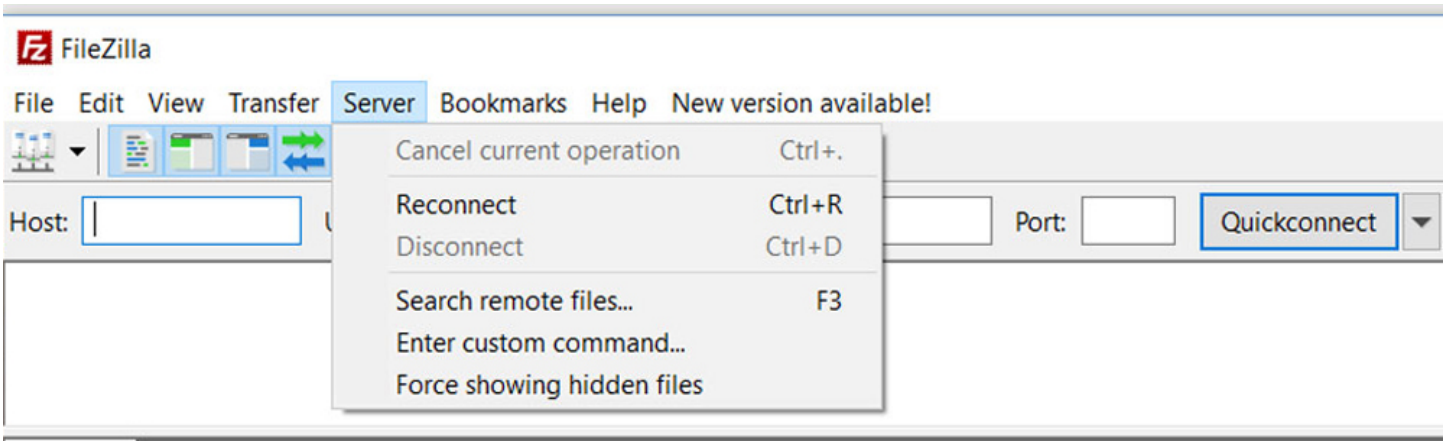
The are numerous free FTP programs available. For safety and ease of usability we recommend FileZilla. You can download a safe, free version of the program for Windows or Mac. Click the "Download" button, choose "FileZilla Client," and "Save File." The file should offer a dialogue box; click "Run." If no box appears, find the file in your "Downloads" folder and double-click. Once the file is installed, you're on your way to accessing your website's root directory.



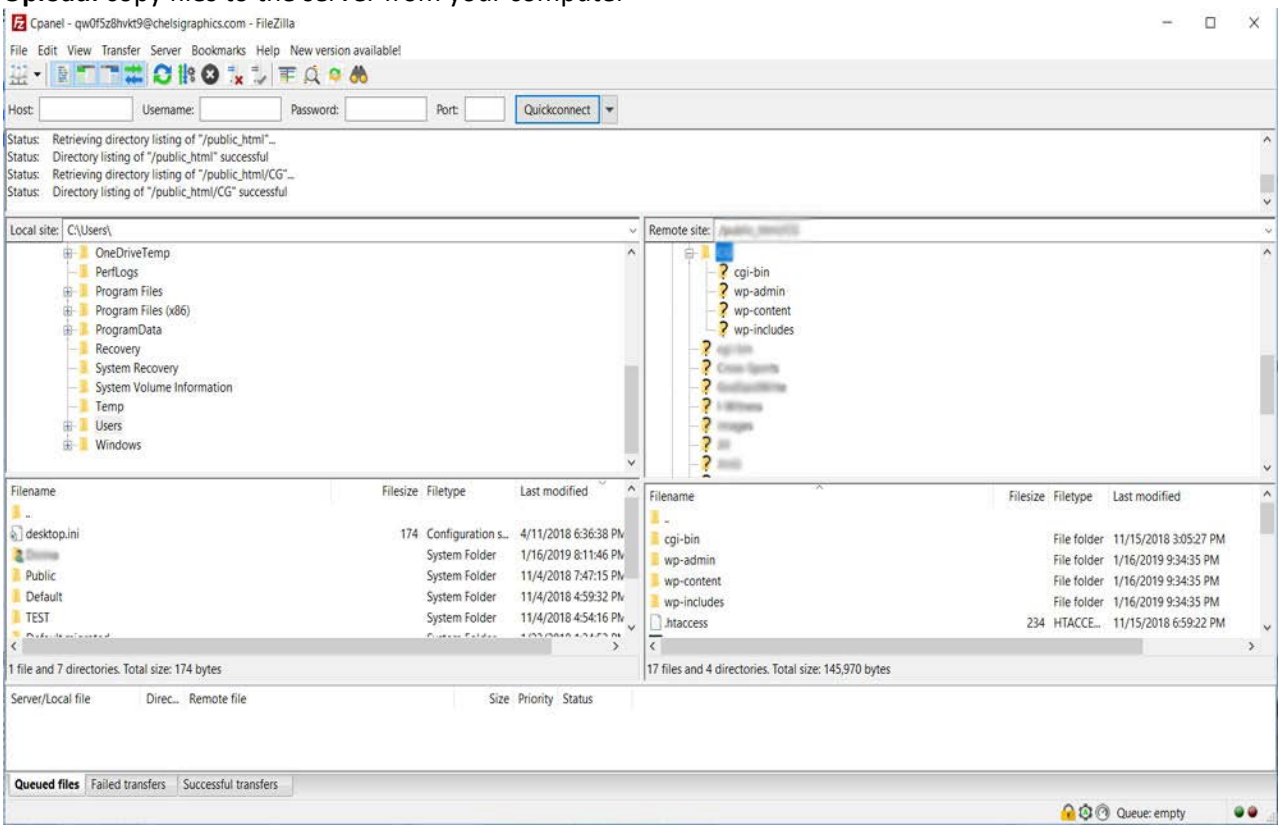
When you open FileZilla, you should see a screen similar to the following image. The FTP window is broken up into a navigation area at the top, followed by four boxes called “quadrants.”

The navigation area includes several empty fields where you will insert your website credentials. If you know how to access your “Hosting Control Panel” you can find this information there. Otherwise, contact your host and request the credentials to access your website files. You will need input for: “Host,” Username,” and “Password.” At times your host will provide a number for the “Port,” but it will most often fill in automatically. Enter the credentials into the provided boxes and click “Quickconnect.”

After establishing this connection, the information is stored for the next visit. Next time merely navigate to the top; click “Server” and on the dropdown menu and click “Reconnect” (shortcut: Ctrl or Cmd + R).




Once connected all four quadrants will contain folders. The left quadrant contains a list of the folders on your computer. The right quadrant now contains a list of the folders on your host server pertaining to your account. The top quadrants will only show folders. The lower quadrants will list both folders and files. The FTP windows will allow you to:
**Download:** copy files from the server to your computer
**Upload:** copy files to the server from your computer



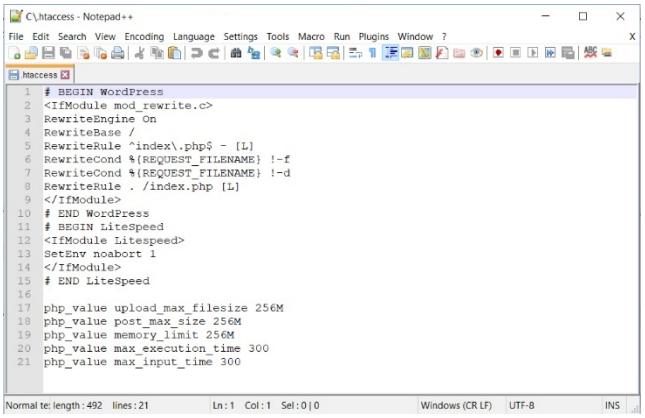
You may have to drill down, depending upon the server technology, possibly through a “public_html” folder to locate your website’s files. Your host can help there, too. WordPress websites will always contain “wp-admin,” “wp-content,” and “wp-includes” folders. The folder where these three folders reside is called the “root directory: and the .htaccess file will also be found there.

To copy the .htaccess file to your computer choose the destination folder in the bottom left quadrant and be sure it is visible. Click and drag the .htaccess file from the lower right quadrant to the destination folder on your computer in the lower left quadrant. A notification will appear confirming the completed transfer.

If the .htaccess file is not visible in your root folder, close the FTP program and return to your WP dashboard. Click “Settings” then “Permalinks.” In this window simply click “Save Changes.” No alterations are necessary. Reopen the FTP and the .htaccess file will now be visible in the lower, right quadrant of your FTP window.

## Step 2: Duplicate File

**Strongly Recommended:** It’s best practice to duplicate the file so that you can leave the original intact, to protect against errors. To do this click the .htaccess file **in the bottom LEFT quadrant** and choose “Open.” The file will open in your computer’s default text editor, as shown below.




Click “File” and “Save as” from the dropdown menu. Save the file as .htaccess-ORIGINAL and click the “Save” button. Now close the text editor. There are now two .htaccess files saved to your computer. This may seem like a tedious, unnecessary step, but just one mistake could break your website or even render it unreachable. By having a copy of the original file, clearly marked, you can change its name back to .htaccess and upload it to restore your website to its original state.

## Step 3. Customizing via .htaccess

Customizing through the .htaccess file offers so many options that we can only cover a few in this article. We’ve chosen a to cover Redirects and Security options here.

### Redirects

There are many times when an automatic redirect to another location is an invaluable tool. In the case of a changed website address (URL), visitors must be sent to the new location. This is also necessary when a single page or blog post has been moved to a new location. These instances require what’s called a “301 Redirect” which will automatically navigate to a new location.

#### Website Redirect

Add the following code (or snippet) to your .htaccess file. Update the “example.com” to your previous URL in both places. Your current URL will replace the “http://example.net/” on the last line. Be sure that you use “https://” if your website is protected by an SSL certificate which is explained in this video link.

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

**Single Page Redirect**
In the case of redirecting a single page, add the following snippet to your .htaccess file:

*Redirect 301 /oldpage.html http://www.yoursite.com/newpage.html*

**Copy the code exactly** including spaces and symbols. Alter the code to your own specifications. The "/oldpage.html" must be changed to the exact name of the page that's moving. Your website must be exactly entered, including an "https:" if you're site has security enabled. Finally, replace the "/newpage.html" with the exact name of the new page's location, including your URL. Upload the edited .htaccess file to your root directory via FTP and refresh your website to test the changes.

## Security

To enhance the security of your website, add these snippets to your .htaccess file. Upload the edited file to the root directory via FTP.

**Protecting essential website files**
Protect from unauthorized users

*<files ~ "^.*\.([Hh][Tt][Aa])">*
*order allow,deny*
*deny from all*
*satisfy all*
*</files>*

Protect your credentials

*<files wp-config.php>*
*order allow,deny*
*deny from all*
*</files>*

Protect "include-only" files which never need to be altered ergo access should be blocked

*<IfModule mod_rewrite.c>*
*RewriteEngine On*
*RewriteBase /*
*RewriteRule ^wp-admin/includes/ - [F,L]*
*RewriteRule !^wp-includes/ - [S=3]*
*RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]*
*RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]*
*RewriteRule ^wp-includes/theme-compat/ - [F,L]*
*</IfModule>*

**Restrict Access to Your Website**
Replace the IP address with that of known spammers

*<Limit GET POST>*
*order allow,deny*

*deny from 123.456.78.9*
*allow from all*
*</Limit>*

**Disable Directory Browsing**
Block unauthorized browsing of your website files

*disable directory browsing*
*Options All -Indexes*

To protect specific folders from hackers, copy the original .htaccess file into the designated folder and add the snippet provided.

The “wp-content” folder contains your design, content, and media; a prime target for hackers
Create a copy of the original .htaccess file into the wp-content folder and add the following snippet.

*Order deny,allow*
*Deny from all*
*<Files ~ ".(xml|css|jpe?g|png|gif|js)$">*
*Allow from all*
*</Files>*

Limit Access to WP Admin Panel: restrict unauthorized logins
Create a copy of the original .htaccess file into the wp-admin folder and add the following snippet. Be sure to update the “12.34.56.78” to your IP address. Locate your IP address by visiting IP Location.

*# Limit logins and admin by IP*
*<Limit GET POST PUT>*
*order deny,allow*
*deny from all*
*allow from 12.34.56.78*
*</Limit>*

## In Closing

There are a myriad of snippets available to enhance your website’s functionality. You can find many more by searching the web. We’ve given you a good start.

**Remember:**

1. Back up your site before doing anything else. We recommend the Updraft Plus plugin.
2. Always keep an unaltered copy of the original .htaccess file.
3. Make one alteration at a time and test before continuing.

Maintaining and securing your website can seem intimidating. It is a learning process but will eventually come more easily. Working with the .htaccess file is a great starting point to learn some basics.

Enjoy the journey.