

Htaccess

💡 THIS IS A SCRATCHPAD ONLY PAGE AT THE MOMENT, AND SHOULD NOT BE MOVED OUT OF HERE 💡

💡 This document should be used to draft a new version of the [official httpd howto docs](#) 💡

💡 Please review/delete <http://wiki.apache.org/general/htaccess> once this page graduates from the ScratchPad 💡

The use of .htaccess files is discouraged as they can have a detrimental effect on server performance. Only use them when necessary.

What is the purpose of .htaccess files?

The purpose of .htaccess files is to provide a means to configure Apache for users who cannot modify the main configuration file (usually httpd.conf; see [DistributionsDefaultLayout](#)).

.htaccess myths

.htaccess files mean password protection

Not really; .htaccess files **can** be used to provide password protection the same way the main configuration files can be used to provide password protection; there is nothing special about .htaccess for this purpose.

It has to be called .htaccess

No, you can define its name through the [AccessFileName](#) directive. It isn't recommended practice, though, especially if you'll provide shared hosting or something similar where uninformed people will have web space and expect .htaccess files to be supported.

When should I, and should I not use .htaccess files?

Allowing .htaccess files will make Apache look for them upon every access to your server. Since parent directories are searched as well, this will take some (small) amount of time, and can impact your server's performance. For a better explanation, see [HtaccessGotchas](#). (move that explanation here?)

Should use

.htaccess files should really only be used when you cannot directly edit the main configuration files.

Should not use

You should not use .htaccess when:

1. you have access to edit the main server configuration file(s).
2. server performance is of concern to you.
3. untrusted people host websites on the server. (See [How can I prevent users from using .htaccess?](#) and [How can I control what users can do with .htaccess files?](#) (How to link to headings of this doc????))

How can I create a new .htaccess file?

Use your favourite editor to create a .htaccess file in the folder where you want it to take effect. Make sure that the file can be read by Apache's UID.

What can I do with my .htaccess file?

.htaccess files are containers for a subset of Apache directives. .htaccess files apply to the directory they are placed in and all its descendants. I.e. a .htaccess file in /path/to/site, will apply to that directory, but to /path/to/site/images too!

This means you can think of .htaccess files as dynamically adding the following to the master Apache configuration:

```
<Directory /path/to/site>
# .htaccess content goes here
</Directory>
```

You must place the .htaccess file in the directory where you want it to take effect. For example if you want to use a .htaccess file to force authentication for <http://www.example.com/admin>, and your DocumentRoot is set to /var/www/html/www.example.com, you would place the .htaccess file in /var/www/html/www.example.com/admin .

So, what can I use?

You can put in a .htaccess file any Apache directive that the administrator chooses to enable, from the subset of directives that Apache supports in .htaccess files. This works by setting the AllowOverride directive correctly.

To check what the administrator enabled, contact them or your hosting provider, or just try.

To check what directives Apache supports in .htaccess files, look for the directive you would like to use in the documentation and check out the [Context](#) section. If it says .htaccess, then that directive is valid in .htaccess, unless it has been disabled by the administrator.

An example. Let's suppose you really must use .htaccess files and that you want to password protect the location we talked about above, located at /var/www/html/www.example.com/admin. A .htaccess file for that task should look like:

/var/www/html/www.example.com/admin/.htaccess

```
AuthType Basic
AuthName "Authentication Required"
AuthUserFile /etc/htpasswd/.htpasswd.example.com
Require valid-user

Order deny,allow
```

How can I control what users can do with .htaccess files?

By setting [AllowOverride](#) in the proper <Directory> directive. See the [AllowOverride](#) documentation.

But, I really dislike having all the directives in one huge config file, and .htaccess files help me with that!

This is not a valid reason, since you can use [Include](#).

(I suggest moving this part to its own section) You can use the Include directive to alleviate that if it really bothers you! Let's see how.

We'll suppose you dislike huge config files and that, for instance, you want to provide password authentication for all your virtual hosts. You could put a .htaccess in the root of each virtual host, or you could create a subdirectory within the Apache config directory, containing a config file per vhost with the proper directives – such as /usr/local/apache2/conf/vhosts-protection. Then, in httpd.conf, you could write

```
Include /usr/local/apache2/conf/vhosts-protection/*.conf
```

/usr/local/apache2/conf/vhosts-protection/www.example.com.conf

```
<Directory /var/www/html/www.example.com>
AuthType Basic
AuthName "Authentication Required"
#Change the following to either a single file for all domains or to a different naming scheme if you like
AuthUserFile /etc/htpasswd/.htpasswd.example1.com
Require valid-user

Order allow,deny
Allow from all
</Directory>
```

And so on for the rest of your virtual hosts. (end of suggest)

How can I prevent users from using .htaccess?

In your main server config, place the following in your top-level <Directory> block. E.g.

```
<Directory /var/www/html>
# ... other directives
AllowOverride None
# ... other directives
</Directory>
```

Remember the application of a <Directory> block is recursive. So if you set it at the top level it will apply to all sub-directories unless explicitly reversed through a different <Directory> directive.

How do I troubleshoot my .htaccess files?

Try putting garbage in it. If it is being read, you'll get an Internal Server Error when accessing that URL. If it is being read, continue to [HtaccessGotchas](#) to find more about what might be wrong.