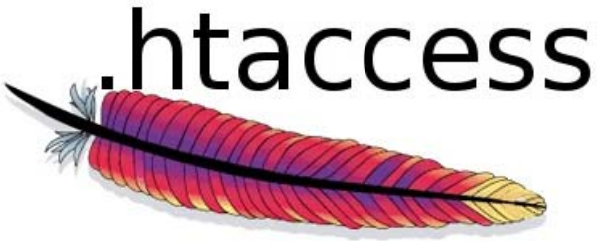
htaccess Giude



Collected by www.tqani.com

What is .htaccess?

.htaccess is a configuration file for use on web servers running the Apache Web Server software. When a .htaccess file is placed in a directory which is in turn 'loaded via the Apache Web Server', then the .htaccess file is detected and executed by the Apache Web Server software. These .htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer. These facilities include basic redirect functionality, for instance if a 404 file not found error occurs, or for more advanced functions such as content password protection or image hot link prevention.

Table of Contents

How to use .htaccess

'.htaccess' is the filename in full, it is not a file extension. For instance, you would not create a file called, 'file.htaccess', it is simply called, '.htaccess'. This file will take effect when placed in any directory which is then in turn loaded via the Apache Web Server software. The file will take effect over the entire directory it is placed in and all files and subdirectories within the specified directory.

You can create a .htaccess file using any good text editor such as TextPad, UltraEdit, Microsoft WordPad and similar (you cannot use Microsoft NotePad).
Here is an example of what you might include in a .htaccess file.

```
AuthName "Member's Area Name"
AuthUserFile /path/to/password/file/.htpasswd
AuthType Basic
require valid-user
ErrorDocument 401 /error_pages/401.html
AddHandler server-parsed .html
```

This is a fairly advanced example: it enables password protection on the directory; it offers redirection to a custom error page if a user fails to login correctly; and it enables SSI (server side includes) for use with '.html' files. Please don't be put off, it's very simple once you gain a basic understanding and this article provides examples which are ready to go - simply copy, paste and customise. Examples are explained line by line so it is clear exactly what each line does and why you need it.

Once you have created a .htaccess file, which may look similar to the one shown above (or may simply contain one line), you need to upload it. This should be done using a FTP (file transfer protocol) program. You should already have one which you will have used to upload your web site content. If not, many are available free of charge from web sites such as 'Download.com' and we can recommend 'CuteFTP' and 'WSFTP'.

When uploading your .htaccess files, it is very important you upload the file in 'ASCII' mode. 'ASCII' and 'BINARY' are different methods of transferring data and it is important .htaccess files are transferred in 'ASCII' mode and not 'BINARY'. It is likely your FTP software will default to 'BINARY' so look for a 'Transfer Mode' or 'Transfer Type' option in the menus.

Upload the .htaccess file to the directory you would like it to take effect over. Now visit this directory using your web browser as you would for any other document on your web site and check it has worked correctly.

Note, when you upload your .htaccess file it may not appear in the directory listings for files on your web site. Do not worry; this means your server or FTP software is hiding them which should not be an issue.

A possible cause of error is if the file permissions on the .htaccess file are not set correctly. This only occurs on certain servers, but you may like to change the permissions on the file to '755' or 'executable'. You can do this with your FTP software, look for a 'File Permissions' or 'CHMOD' option, and input '0755'.

If your .htaccess file does not work, you should contact your system administrator or web hosting company and ensure they have enabled .htaccess within your account. Some web hosting companies do not allow use without permission. If errors persist, consult this article for advice, or contact your system administrator for advice.

Error documents

Creating custom error pages is very useful, it allows you to show web site visitors a friendly error message, for instance if a URL on your web site does not work. This avoids the unfriendly '404 File Not Found' error and allows you to display a friendly error, explaining possible solutions and guiding the visitor back into your web site content, rather than leaving them frustrated and lost.

To set-up custom error documents, create a .htaccess file following the main instructions and guidance which includes the following text:

```
ErrorDocument 404 /error_pages/404.html
```

The above line tells the Apache Web Server to display the document located at /error_pages/404.html (under your domain name/web site address) whenever a 404 (file not found) error occurs.

In this example, we have assumed you have created the error document and called it '404.html' and that you have placed it in a directory called 'error_pages' under your domain name. For example, http://www.yourdomain.com/error_pages/404.html

The document 404.html is a normal HTML document like the others in your web site and can display whatever content you wish, however we recommend including a 'File Not Found' message.

To setup further error documents, for example for '401 Unauthorised', '403 Forbidden', and '500 Internal Server' error messages, create a .htaccess file following the main instructions and guidance which includes the following text:

```
ErrorDocument 401 /error_pages/401.html
ErrorDocument 404 /error_pages/404.html
ErrorDocument 500 /error_pages/500.html
```

It's all very well displaying a friendly error message, but more importantly you need to resolve the error. By using a CGI script instead of a static HTML document as the error document allows us to record errors in a database, and resolve them.

This can be achieved very easily thanks to a variety of pre-made solutions which can even show us which errors are received most frequently. Such products can be found on The CGI Resource Index and HotScripts.com.

Redirects

Redirects enable us to direct web site visitors from one document within your web site to another. This is useful for example, if you have moved your web site content and would like to redirect visitors from old links to the new content location.

To set-up redirects, create a .htaccess file following the main instructions and guidance which includes the following text:

> *Redirect /old_dir/ http://www.yourdomain.com/new_dir/index.html*

The above line tells the Apache Web Server that if a visitor requests a documents located in the directory 'old_dir', then to display the document 'index.html' located in the directory 'new_dir'.

You see in this example, the 'old_dir' is the location of the document to be requested by the visitor, and is a document or directory located under your main domain. In this example, the directory 'old_dir' would be located at 'http://www.yourdomain.com/old_dir/'. However, you will also notice the location of the file that the visitor is to be redirected to is a full web site URL, not what is referred to as a relative URL in the case of 'old_dir'. This means we can redirect visitors to the 'old_dir' folder to any web site document, it doesn't have to be held within your web site content and could be any web site.

It is very important (and the most common cause of error) that you understand the difference between a relative URL and an absolute/full URL. A relative URL is the location of the document within the web site, and does not include the actual domain name of the web site. These are used for documents held within the web site to simplify and shorten the URL. A absolute or full URL is one which includes the full domain name.

For example, for a absolute/full URL, 'http://www.yourdomain.com/directory/file.html'. the relative URL for this document would be, '/directory/file.html'.

Password protection

The password protection and authentication systems offered by the **Apache Web Server** are probably the most important use of .htaccess files. Very easily, we can password protect a directory (or multiple) of a web site which require a username and password to access. The login procedure for these secure directories is handled automatically by the web browser using a pop-up login interface (you've probably seen these before). Passwords are also encrypted using one of the best encryption methods available which ensures login credentials are kept secure. In this section we will discuss the details of the .htaccess authentication system, we will explain how to set-up password protection, and a variety of helpful related information, we will also explain a variety of pre-made software which can be used to accomplish these tasks.

To begin, decide which directory you would like to password protect (note that all files and subdirectories within the directory will be password protected), then create a .htaccess file following the main instructions and guidance which includes the following text:

```
AuthName "Member's Area Name"
AuthUserFile /path/to/password/file/.htpasswd
AuthType Basic
require valid-user
```

The first line tells the Apache Web Server the secure directory is called 'Member's Area Name', this will be displayed when the pop-up login prompt appears. The second line specifies the location of the password file. The third line specifies the authentication type, in this example we are using 'Basic' because we are using basic HTTP authentication and finally the fourth line specifies that we require valid login credentials, this line can also be used to specify a specific username, e.g. 'require user username' would require the username 'username'. You would use this if you were password protecting an administration area, rather than setting up a public password protected directory.

The location of the password file can be anywhere on your web server, the '/location/of/password/file/' must be replaced with the full/absolute path to the directory containing the password file, and the '.htpasswd' file must exist, this can however be called anything. We use the filename '.htpasswd' because the server will recognise the filename and will hide it from visitors. Note, some servers do require the password file be located in the same directory as the .htaccess file. It is also important to use a full/absolute server path for the location of the password file, a relative path, or any variation of a URL will not work.

The password file would contain something similar to the following text:

```
username:encryptedpassword
fred_smith:oCF9Pam/MXJg2
```

Now, you cannot just make up the password, on Unix/Linux servers they must be encrypted by the server, on Windows servers you do just use a plain text password as Windows does not offer any encryption methods. You can have any number of user records in your password file, one account per row, separating the username and password with a colon. **ionix**, offer a product called '**DirectoryPass**' which is very useful for this task. DirectoryPass is a very simple Perl script which automates the process of setting up password protection, and more importantly creating the user accounts.

Setting up a password protected directory allows you to offer a member's area, offering a member's area on your web site is also a great way of tracking your web site visitors and a brilliant way of building a community feel on the web site. By asking visitors to register to access the content you are able to collect whatever information about the visitors that you require, whether it be the visitors country of residence, sex or professional status.

Such a system can be setup very easily these days thanks to the vast array of pre-made solutions available on the Internet, most of which are as easy to setup as the initial web site content. ionix offers two other solutions to this scenario which have proven very popular, one is called '**Locked Area**' which has been available on the Internet for over eight years and is now in use on over fifty thousand web sites. This a very simple but effective member's area management system which is available completely free of charge, it can be used to set-up a secure member's area to store your content, it includes a registration system for your web site visitors to register for member's area access, and it includes a fantastic administration panel to help you manage accounts and email your members.

Another product offered by ionix is, '**OpenCrypt**'. 'OpenCrypt' is ionix's enterprise solution for membership management and offers some impressive facilities including an extensive statistical analysis system and possibly the most flexible registration system seen in this market. Both products manage the security of your content and ensure that visitors cannot access the member's area without registering, 'OpenCrypt' also offers comprehensive facilities to prevent your visitors from sharing their login details which is especially important with high-demand web sites.

A variety of other solutions are available to accomplish this task, our advice would be to use a piece of software written in preferably Perl, or alternatively PHP. In our experience we would not recommend the use of software written in ASP or ColdFusion particularly when looking for security related software. We have listed a number of web sites which you will find useful when looking for similar solutions, namely **Hot Scripts**, **CGI Resources** and **The CGI-Index.com**.

Note, it is not possible to offer a log-out facility, the login details are cached in the web browser until the browser is closed, so visitors may leave the web site and return later in the session without being prompted to login again. When the browser is closed and re-opened the login details are deleted from the cache and the pop-up prompt will be displayed. The log-out facility has been discussed for some time, various methods have been suggested but none are reliable enough to be worth discussing.

Deny visitors by IP address

The visitor blocking facilities offered by the Apache Web Server enable us to deny access to specific visitors, or allow access to specific visitors. This is extremely useful for blocking unwanted visitors, or to only allow the web site owner access to certain sections of the web site, such as an administration area.

To set-up visitors restrictions and blocking, create a .htaccess file following the main instructions and guidance which includes the following text:

```
order allow,deny
deny from 255.0.0.0
deny from 123.45.6.
allow from all
```

The above lines tell the Apache Web Server to block visitors from the IP address '255.0.0.0' and '123.45.6.', note the second IP address is missing the fourth set of digits, this means any IP address which matches the firth three set of digits will be blocked, e.g. '123.45.6.10' and '123.45.6.255' would be blocked.

To set-up blocking of all visitors except yourself, create a .htaccess file following the main instructions and guidance which includes the following text:

```
order allow,deny
allow from 255.0.0.0
deny from all
```

The above lines tell the Apache Web Server to block all visitors except those with the IP address '255.0.0.0', which you should replace with your own IP address.

You may add any number of 'deny from' and 'allow from' records after the 'order allow,deny'. Note the change from 'allow from all' to 'deny from all' on the bottom line, this is important and must be changed depending on your requirements. If you want to allow your visitor access, you would use 'allow from all' and place 'deny from' lines above.

Blocked visitors will be shown a '403 Forbidden' error message. You can customise this error message by following the 'Error Documents' section of this article.

Deny visitors by referrer

The visitor blocking facilities offered by the Apache Web Server enable us to deny access to specific visitors based on where they have come from. If you've ever looked at your logs and noticed a surprising increase in traffic, yet no increases in actual file requests it's probably someone pinching content (such as CSS files) or someone attempting to hack your web site (this may simply mean trying to find non public content).

Note, this functionality requires that 'mod_rewrite' is enabled on your server. Due to the demands that can be placed on system resources, it is unlikely it is enabled so be sure to check with your system administrator or web hosting company.

To set-up block a single referrer, create a .htaccess file following the main instructions and guidance which includes the following text:

```
RewriteEngine on
# Options +FollowSymlinks
RewriteCond %{HTTP_REFERER} otherdomain\.com [NC]
RewriteRule .* - [F]
```

The above lines tell the Apache Web Server to block traffic from the URL 'otherdomain.com'. The '[NC]' text after the referrer specifies it as not case-sensitive. Which prevents traffic from 'OtherDomain.com', 'otherdomain.com', 'OTHERDOMAIN.COM' and so on.

To set-up block multiple referrers, create a .htaccess file following the main instructions and guidance which includes the following text:

```
RewriteEngine on
# Options +FollowSymlinks
RewriteCond %{HTTP_REFERER} otherdomain\.com [NC,OR]
RewriteCond %{HTTP_REFERER} anotherdomain\.com
RewriteRule .* - [F]
```

The above lines tell the Apache Web Server to block traffic from the URL 'otherdomain.com' and 'anotherdomain.com'. Note the backslash before the dot, this is important, e.g. 'domain\.com'. The only difference between blocking a single referrer and multiple referrers is the modified [NC, OR] flag in the multiple referrers example, this should be added to every domain except the last.

You might have noticed the line "Options +FollowSymlinks" above, which is commented with a '#'. Uncomment this line if your server returns a '500 Internal Server' error. This means your server isn't configured with FollowSymLinks in the '' section of the 'httpd.conf'. Contact your system administrator for advice with this issue.

Blocked referrers will be shown a '403 Forbidden' error message. You can customise this error message by following the 'Error Documents' section of this article.

Hot link prevention techniques

Hot link prevention refers to stopping web sites that are not your own from displaying your files or content, e.g. stopping visitors from other web sites. This is most commonly used to prevent other web sites from displaying your images but it can be used to prevent people using your JavaScript or CSS (cascading style sheet) files. The problem with hot linking is it uses your bandwidth, which in turn costs money, hot linking is often referred to as 'bandwidth theft'.

Using .htaccess we can prevent other web sites from sourcing your content, and can even display different content in turn. For example, it is common to display what is referred to as an 'angry man' images instead of the desired images.

Note, this functionality requires that 'mod_rewrite' is enabled on your server. Due to the demands that can be placed on system resources, it is unlikely it is enabled so be sure to check with your system administrator or web hosting company.

To set-up hot link prevention for '.gif', '.jpg' and '.css' files, create a .htaccess file following the main instructions and guidance which includes the following text:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteRule \.(gif|jpg|css)$ - [F]
```

The above lines tell the Apache Web Server to block all links to '.gif', '.jpg' and '.css' files which are not from the domain name 'http://www.yourdomain.com/'. Before uploading your .htaccess file ensure you replace 'yourdomain.com' with the appropriate web site address.

To set-up hot link prevention for '.gif', '.jpg' files which displays alternate content (such as an angry man image), create a .htaccess file following the main instructions and guidance which includes the following text:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?yourdomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ http://www.yourdomain.com/hotlink.jpg [R,L]
```

The above lines tell the Apache Web Server to block all links to '.gif' and '.jpg' files which are not from the domain name 'http://www.yourdomain.com/' and to display the file 'http://www.yourdomain.com/hotlink.jpg' instead. Before uploading your .htaccess file ensure you replace 'yourdomain.com' with the appropriate web site address.

Blocking offline browsers and 'bad bots'

Offline browsers are pieces of software which download your web page, following the links to your other web pages, downloading all the content and images. The purpose of this is innocent, so the visitor can log off the Internet and browse the site without a connection, but the demand on the server and bandwidth usage can be expensive. Bad bots as they are often called refers to programs which visit your web site, either to source content, look for security holes or to scan for email addresses. This is often how your email address ends up on 'Spam' databases, because they have set a 'bot' to scan the Internet and collect email addresses. These programs and 'bots' often ignore the rules set out in 'robot.txt' files.

Below is a useful example of how to block some common 'bots' and site rippers. Create a .htaccess file following the main instructions and guidance which includes the following text:

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^BlackWidow [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bot\ mailto:craftbot@yahoo.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^ChinaClaw [OR]
RewriteCond %{HTTP_USER_AGENT} ^Custo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCo [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Demon [OR]
RewriteCond %{HTTP_USER_AGENT} ^eCatch [OR]
RewriteCond %{HTTP_USER_AGENT} ^EirGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR]
RewriteCond %{HTTP_USER_AGENT} ^Express\ WebPictures [OR]
RewriteCond %{HTTP_USER_AGENT} ^ExtractorPro [OR]
RewriteCond %{HTTP_USER_AGENT} ^EyeNetIE [OR]
RewriteCond %{HTTP_USER_AGENT} ^FlashGet [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetRight [OR]
RewriteCond %{HTTP_USER_AGENT} ^GetWeb! [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go!Zilla [OR]
RewriteCond %{HTTP_USER_AGENT} ^Go-Ahead-Got-It [OR]
RewriteCond %{HTTP_USER_AGENT} ^GrabNet [OR]
RewriteCond %{HTTP_USER_AGENT} ^Grafula [OR]
RewriteCond %{HTTP_USER_AGENT} ^HMView [OR]
RewriteCond %{HTTP_USER_AGENT} HTTrack [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Stripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^Image\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} Indy\ Library [NC,OR]
RewriteCond %{HTTP_USER_AGENT} ^InterGET [OR]
RewriteCond %{HTTP_USER_AGENT} ^Internet\ Ninja [OR]
RewriteCond %{HTTP_USER_AGENT} ^JetCar [OR]
RewriteCond %{HTTP_USER_AGENT} ^JOC\ Web\ Spider [OR]
RewriteCond %{HTTP_USER_AGENT} ^larbin [OR]
RewriteCond %{HTTP_USER_AGENT} ^LeechFTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mass\ Downloader [OR]
RewriteCond %{HTTP_USER_AGENT} ^MIDown\ tool [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mister\ PiX [OR]
RewriteCond %{HTTP_USER_AGENT} ^Navroad [OR]
RewriteCond %{HTTP_USER_AGENT} ^NearSite [OR]
```

```
RewriteCond %{HTTP_USER_AGENT} ^NetAnts [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Net\ Vampire [OR]
RewriteCond %{HTTP_USER_AGENT} ^NetZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Octopus [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Explorer [OR]
RewriteCond %{HTTP_USER_AGENT} ^Offline\ Navigator [OR]
RewriteCond %{HTTP_USER_AGENT} ^PageGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^Papa\ Foto [OR]
RewriteCond %{HTTP_USER_AGENT} ^pavuk [OR]
RewriteCond %{HTTP_USER_AGENT} ^pcBrowser [OR]
RewriteCond %{HTTP_USER_AGENT} ^RealDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^ReGet [OR]
RewriteCond %{HTTP_USER_AGENT} ^SiteSnagger [OR]
RewriteCond %{HTTP_USER_AGENT} ^SmartDownload [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperBot [OR]
RewriteCond %{HTTP_USER_AGENT} ^SuperHTTP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Surfbot [OR]
RewriteCond %{HTTP_USER_AGENT} ^tAkeOut [OR]
RewriteCond %{HTTP_USER_AGENT} ^Teleport\ Pro [OR]
RewriteCond %{HTTP_USER_AGENT} ^VoidEYE [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Image\ Collector [OR]
RewriteCond %{HTTP_USER_AGENT} ^Web\ Sucker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebAuto [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebCopier [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebFetch [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebGo\ IS [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebLeacher [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebReaper [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebSauger [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ eXtractor [OR]
RewriteCond %{HTTP_USER_AGENT} ^Website\ Quester [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebStripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebWhacker [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Wget [OR]
RewriteCond %{HTTP_USER_AGENT} ^Widow [OR]
RewriteCond %{HTTP_USER_AGENT} ^WWWOFFLE [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xaldon\ WebSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus
RewriteRule ^.* - [F,L]
```

DirectoryIndex uses

The directoryindex command allows you to specify a default page to display when a directory is accessed. For instance, if a visitor requests a directory on your web site, you can specify the file to load when the directory is accessed (if a filename is not specified in the initial request). For example, to display a 'index.html' file rather than showing directory listings or to load a 'index.php' file rather than an 'index.html' file.

To set-up a directoryindex, create a .htaccess file following the main instructions and guidance which includes the following text:

```
DirectoryIndex index.html
```

The above lines tell the Apache Web Server to display the 'index.html' file, whenever the directory containing this .htaccess file (or any subdirectory) is accessed.

We can setup a directoryindex to call multiple files using the following text:

```
DirectoryIndex index.html index.cgi index.php
```

The above lines tell the Apache Web Server to display the 'index.html' file as the directoryindex, if this file is not available then display 'index.cgi', and if this is not available then display 'index.php'.

If not of the specified files are available, the Apache Web Server will revert to it's default settings, either displaying an error message, a directory listings not available message, or displaying the directory listings of files and directories (this can be prevented which we discuss in section 'Prevent viewing of directory listings').

Adding MIME types

MIME types set what a file is, or rather what file extensions refer to what file types. For example, a '.html' file extension refers to a HTML document, a '.zip' file extension refers to a ZIP archive file. The server needs to know this so it knows how to deal with the file. This is often used to create custom file extension for common file types.

To setup a MIME type, create a .htaccess file following the main instructions and guidance which includes the following text:

```
AddType text/html htm0
```

'AddType' specifies that you are adding a MIME type. The second part is the MIME type, in this case text or HTML, and the final part is the file extension, in this example 'htm0'.

A common issue with MP3 or SWF files not playing can be resolved with the following text:

```
AddType application/x-shockwave-flash swf
```

A handy trick, to force a file to be downloaded, via the 'Save As' feature in the web browser, set the MIME type to application/octet-stream and the browser will immediately prompt for download. Note, this does not work consistently in some versions of Microsoft Internet Explorer.

Here is a list of various MIME types and some associations:

```
AddType text/html .html .htm
AddType text/plain .txt
AddType text/richtext .rtx
AddType text/tab-separated-values .tsv
AddType text/x-setext .etx
AddType text/x-server-parsed-html .shtml .sht
AddType application/macbinhex-40 .hqx
AddType application/netalivelink .nel
AddType application/netalive .net
AddType application/news-message-id
AddType application/news-transmission
AddType application/octet-stream .bin .exe
AddType application/oda .oda
AddType application/pdf .pdf
AddType application/postscript .ai .eps .ps
AddType application/remote-printing
AddType application/rtf .rtf
AddType application/slate
AddType application/zip .zip
AddType application/x-mif .mif
AddType application/wita
AddType application/wordperfect5.1
AddType application/x-csh .csh
AddType application/x-dvi .dvi
AddType application/x-hdf .hdf
AddType application/x-latex .latex
AddType application/x-netcdf .nc .cdf
```

AddType application/x-sh .sh
AddType application/x-tcl .tcl
AddType application/x-tex .tex
AddType application/x-texinfo .texinfo .texi
AddType application/x-troff .t .tr .roff
AddType application/x-troff-man .man
AddType application/x-troff-me .me
AddType application/x-troff-ms .ms
AddType application/x-wais-source .src
AddType application/x-bcpio .bcpio
AddType application/x-cpio .cpio
AddType application/x-gtar .gtar
AddType application/x-shar .shar
AddType application/x-sv4cpio .sv4cpio
AddType application/x-sv4crc .sv4crc
AddType application/x-tar .tar
AddType application/x-ustar .ustar
AddType application/x-director .dcr
AddType application/x-director .dir
AddType application/x-director .dxr
AddType application/x-onlive .sds
AddType application/x-httpd-cgi .cgi
AddType image/gif .gif .GIF
AddType image/ief .ief
AddType image/jpeg .jpeg .jpg .jpe .JPG
AddType image/tiff .tiff .tif
AddType image/x-cmu-raster .ras
AddType image/x-portable-anymap .pnm
AddType image/x-portable-bitmap .pbm
AddType image/x-portable-graymap .pgm
AddType image/x-portable-pixmap .ppm
AddType image/x-rgb .rgb
AddType image/x-xbitmap .xbm
AddType image/x-xpixmap .xpm
AddType image/x-xwindowdump .xwd
AddType audio/basic .au .snd
AddType audio/x-aiff .aif .aiff .aifc
AddType audio/x-wav .wav
AddType audio/x-pn-realaudio .ram
AddType audio/x-midi .mid
AddType video/mpeg .mpeg .mpg .mpe
AddType video/quicktime .qt .mov
AddType video/x-msvideo .avi
AddType video/x-sgi-movie .movie
AddType message/external-body
AddType message/news
AddType message/partial
AddType message/rfc822
AddType multipart/alternative
AddType multipart/appledouble
AddType multipart/digest
AddType multipart/mixed
AddType multipart/parallel
AddType x-world/x-vrml .wrl

Enable SSI with .htaccess

SSI stands for server side includes, these are special HTML tags which you can include in your HTML documents to call CGI scripts or other HTML content. This is particularly useful, for example to include a navigation menu in your HTML documents, it allows you to use one document to display the navigation menu in all your other documents. This saves disk space and means if you need to update the content, you only need to modify one file.

Two examples of HTML tags you would use to call SSI documents are shown below, these would be placed in your HTML document:

```
<!--#exec cgi="/cgi-bin/script.cgi"-->
```

This would load the CGI script 'script.cgi' which is located in the 'cgi-bin' directory.

```
<!--#include virtual="/files/document.html"-->
```

This example would call the HTML document 'document.html' which is located in the 'files' directory. It is important to use a relative URL, not a path or full URL.

It is likely SSI will work on your web server, but you will probably need to use '.shtml' file extensions rather than '.html'. This can be frustrating if you already have a web site setup which uses '.html' extensions. In this case, you can enable SSI by following the instructions below.

To enable SSI, create a .htaccess file following the main instructions and guidance which includes the following text:

```
AddHandler server-parsed .html
```

The above lines tell the Apache Web Server to allow server side includes in documents with the file extension '.html'.

To enable SSI for multiple file extensions, create a .htaccess file following the main instructions and guidance which includes the following text:

```
AddHandler server-parsed .html
AddHandler server-parsed .shtml
AddHandler server-parsed .htm
```

The above lines tell the Apache Web Server to allow server side includes in documents with the file extension '.html', '.shtml' and '.htm'.

Enable CGI outside of the cgi-bin

If your web server does not allow you to run CGI scripts outside of the 'cgi-bin' directory, you can enable CGI. Check with your system administrator or web hosting company before doing so.

To enable CGI, create a .htaccess file following the main instructions and guidance which includes the following text:

```
AddHandler cgi-script .cgi
Options +ExecCGI
```

The above lines tell the Apache Web Server to allow firstly, process '.cgi' files as CGI scripts, and secondly to enable CGI within the directory.

Disable directory listings

Preventing directory listings can be very useful if for example, you have a directory containing important '.zip' archive files or to prevent viewing of your image directories. Alternatively it can also be useful to enable directory listings if they are not available on your server, for example if you wish to display directory listings of your important '.zip' files.

To prevent directory listings, create a .htaccess file following the main instructions and guidance which includes the following text:

```
IndexIgnore *
```

The above lines tell the Apache Web Server to prevent directory listings of directories and files within the directory containing the .htaccess file. The '*' represents a wildcard, this means it will not display any files. It is possible to prevent listings of only certain file types, so for example you can show listings of '.html' files but not your '.zip' files.

To prevent listing '.zip' files, create a .htaccess file following the main instructions and guidance which includes the following text:

```
IndexIgnore *.zip
```

The above line tells the Apache Web Server to list all files except those that end with '.zip'.

To prevent listing multiple file types, create a .htaccess file following the main instructions and guidance which includes the following text:

```
IndexIgnore *.zip *.jpg *.gif
```

The above line tells the Apache Web Server to list all files except those that end with '.zip', '.jpg' or '.gif'.

Alternatively, if your server does not allow directory listings and you would like to enable them, create a .htaccess file following the main instructions and guidance which includes the following text:

```
Options +Indexes
```

The above line tells the Apache Web Server to enable directory listing within the directory containing this .htaccess file. You can also reverse this to disable directory listings by replacing the plus sign before the text 'Indexes' with a minus sign. e.g. 'Options -Indexes'.

You can also include a default description for the directory listings that is displayed at the top of the page by placing a file called 'HEADER' in the same directory. The contents of this file are displayed before the list of directory contents. You can also include a footer, by creating a file called 'README'. The contents of this file are displayed after the list of directory contents.

Setting server timezone

To set your web servers date timezone, for example for Eastern Standard Time (EST) use the following code:

```
SetEnv TZ America/Indianapolis
```

For example, for Los Angeles time (Pacific time), use the following code:

```
SetEnv TZ America/Los_Angeles
```

Other location examples include:

```
America/New_York - Eastern Time
America/Detroit - Eastern Time - Michigan (most locations)
America/Louisville - Eastern Time (Louisville, Kentucky)
America/Indianapolis - Eastern Standard Time (Indiana, most locations)
America/Indiana/Marengo - Eastern Standard Time (Indiana, Crawford County)
America/Indiana/Knox - Eastern Standard Time (Indiana, Starke County)
America/Indiana/Vevay - Eastern Standard Time (Indiana, Switzerland County)
America/Chicago - Central Time
America/Menominee - Central Time (Michigan, Wisconsin border)
America/Denver - Mountain Time
America/Boise - Mountain Time (South Idaho, East Oregon)
America/Shiprock - Mountain Time (Navajo)
America/Phoenix - Mountain Standard Time (Arizona)
America/Los_Angeles - Pacific Time
America/Anchorage - Alaska Time
America/Juneau - Alaska Time (Alaska panhandle)
America/Yakutat - Alaska Time (Alaska panhandle neck)
America/Nome - Alaska Time (west Alaska)
America/Adak - Aleutian Islands
Pacific/Honolulu - Hawaii
```

For European and other timezones it is best to contact the system administrator for further instruction.

Changing server signature

To change the server signature which is displayed as part of the default Apache error documents, use the following code:

```
ServerSignature EMail
SetEnv SERVER_ADMIN nospace@pleasenospace.com
```

The example above will simply change the email address which is displayed, this is useful if the default address is not set correctly.

To remove the server signature completely, use the following code:

```
ServerSignature Off
```

Preventing access to your PHP includes files

If you have a directory containing PHP includes, that you do not wish to be accessed directly from the browser, there is a way of disabling the directory using Mod_Rewrite.

To enable this, create a .htaccess file following the main instructions and guidance, and include the following text:

```
## Enable Mod Rewrite, this is only required once in each .htaccess file
RewriteEngine On
RewriteBase /
## Test for access to includes directory
RewriteCond %{THE_REQUEST} ^[A-Z]{3,9}\ /includes/ .*$ [NC]
## Test that file requested has php extension
RewriteCond %{REQUEST_FILENAME} ^.+\.php$
## Forbid Access
RewriteRule .* - [F,NS,L]
```

Where /includes/ is your includes directory.

Prevent access to php.ini

If you run the risk of someone accessing your php.ini or php.cgi files directly through their browsers, you can limit access to them using .htaccess.

To enable this, create a .htaccess file following the main instructions and guidance, and include the following text:

```
<FilesMatch "^php5?\.(ini|cgi)$">
Order Deny,Allow
Deny from All
Allow from env=REDIRECT_STATUS
</FilesMatch>
```

Forcing scripts to display as source code

If you need to display scripts as source code, instead of executing, for example to allow review, this can be achieved with the Remove Handler function

To enable this, create a .htaccess file following the main instructions and guidance, and include the following text:

```
RemoveHandler cgi-script .pl .cgi .php .py
AddType text/plain .pl .cgi .php .py
```

Ensuring media files are downloaded instead of played

It is possible to ensure that any media files are treated as a download, rather than to be played by the browser.

To enable this, create a .htaccess file following the main instructions and guidance, and include the following text:

```
AddType application/octet-stream .zip .mp3 .mp4
```

This tells the Apache Web Server to treat .zip, .mp3, and .mp4 files as downloadable, and should be used instead of specifiying them as audio/video/zip files in your MIME types section.

Setting up Associations for Encoded Files

Some browsers are capable of uncompressing encoded information as they receive it.

To enable a client to see that a file is encoded, create a .htaccess file following the main instructions and guidance, and include the following text:

```
AddEncoding x-gzip .gz .tgz
AddEncoding x-compress .Z
```

This tells the Apache Web Server to treat .gz and .tgz files as encoded by x-gzip, and .Z files as encoded by x-compress.

Preventing requests with invalid characters

If you wish, you can use Mod_Rewrite to deny requests containing invalid characters, please be aware that with certain site setups this may break links.

To enable this, create a .htaccess file following the main instructions and guidance, and include the following text:

```
RewriteEngine On
RewriteBase /
RewriteCond %{THE_REQUEST} !^[A-Z]{3,9}\ [a-zA-Z0-9\.\+_/\-\?\=\&]+\ HTTP/ [NC]
RewriteRule .* - [F,NS,L]
```

Useful Resources

We have compiled a small list of links to associated resources and web sites.

Apache Resources

- **Apache Tutorial: .htaccess Files** - Official Apache documentation and guidelines.
- **Apache Directives** - A list of directives available in the standard Apache distribution.
- **Apache Documentation** - Main Apache Web Server documentation.

Password Protection Resources

- **HotScripts.com** - User management resources.
- **The CGI Resource Index** - Password protection resources.
- **CGI-Index.com** - Security resources.

You may also be interested in these products, all of which use .htaccess:

- **DirectoryPass** - DirectoryPass is a very powerful, yet simple to use .htaccess management system.
- **Locked Area** - Locked Area is a highly sophisticated password protection and membership management system written in Perl.
- **OpenCrypt** - OpenCrypt is a fully automated and self-managing membership/user management system which is more than capable of the most complex multi-domain installations, whilst still being usable in the most simple of circumstances.

Main Source: http://www.htaccess-guide.com/