



Ultimate htaccess Article

Heres my list of the ultimate htaccess code snippets and examples that I use all the time. I tried to keep them extremely minimalistic.

Don't miss checking out my [example htaccess file](#), its very comprehensive

Hey [jason](#) I'm glad someone finally noticed the "development" htaccess at the bottom of my file! Thats honestly the piece of code I am most glad to have.

Heres the actual code that I use when I'm developing sites for clients

This lets google crawl the page, lets me access the whole site (24.205.23.222) without a password, and lets my client access the page WITH a password. It also allows for XHTML and CSS validation! (w3.org)

```
# HTACCESS FOR WEBDEVELOPERS
#####

# Allow google and developers IP access without pw
# everyone else is prompted for password
AuthName "SiteName Administration"
AuthUserFile /home/sitename.com/.htpasswd
AuthType basic
Require valid-user

Order deny,allow
Deny from all

Allow from 24\.205\.23\.222

Allow from w3.org

Allow from htmlhelp.com
```

Popular Posts

- [Ultimate htaccess Article](#)
- [Sniffing On Ethernet Undetected](#)
- [Vulnerability Scanners Review](#)
- [Speed Up Sites with htaccess Caching](#)
- [Changing Any Password On Xp](#)
- [Ultimate htaccess File sample](#)
- [Apache SSL in htaccess examples](#)
- [Install multiple OS Without Cds](#)
- [Setting charset in htaccess](#)
- [Custom PHP.ini tips and tricks](#)
- [Best htaccess tutorials and articles](#)
- [Using FilesMatch and Files in htaccess](#)
- [SEO in Wordpress](#)
- [PHPMailer tutorial](#)
- [Bypassing VLAN](#)
- [Arp Stuff](#)
- [SPF Records on DreamHost](#)
- [My favorite Javascript 2007](#)
- [COMPUTER SECURITY TOOLBOX](#)
- [FastCGI on DreamHost](#)
- [Custom Boot Menu in Windows XP](#)
- [Delete extra wordpress files Post-Install](#)

Recent Posts

- [Port Redirector](#)
- [Use netcat to create a wrapper path for Nessus](#)

```

Allow from googlebot.com

Allow from google.com

Allow from google-analytics.com

Satisfy Any

# ABSOLUTELY NO CACHING

<FilesMatch ".(html|htm|js|css)$">

Header set Cache-Control "max-age=0, no-cache, no-store, must-revalidate, proxy-revalidate, no-transform, private"

Header set Pragma "no-cache"

FileETag None

ExpiresActive Off

</FilesMatch>

```

Additional and detailed info on each htaccess code snippet can be found at [htaccessElite](#)

NOTE: Most of these snippets can be used with a [Files](#) or [Filesmatch](#) directive to only apply to certain files.

NOTE: Most of the time .htaccess rewrite examples should begin with:

```

Options +FollowSymLinks

RewriteEngine On

RewriteBase /

```

Apache Documentation: 1.3 | 2.0 | 2.2 | Current

Get rid of IE's imagetoolbar

```

<FilesMatch ".(html|htm)$">

Header set imagetoolbar "no"

</FilesMatch>

```

Add Privacy (P3P) Header to your site

To add a [P3P](#) header to your site, which is a good idea, do this.

```
Header set P3P "policyref=\"/w3c/p3p.xml\", CP=\"NOI DSP COR NID CUR ADM DEV OUR BUS\""
```

Learning from the web

- Want to know how to really hack?
- Racing the XP ACL demon
- Top 5 best Vulnerability Port scanners?
- Sniff HTTP to debug Apache
- Shell Scripts for Dreamhost
- Debug apache log files from php
- Redirecting Wordpress Feeds to Feedburner
- Custom PHP.ini tips and tricks
- Custom Boot Menu in Windows XP
- My favorite Javascript 2007

Google Likes

-

Recent Comments

- **bish:** Have you overlooked simple xinetd? service...
- **Google "x-gzip-ok":** They talk about it at...
- **kishfellow:** My view is that many...
- **Cd-MaN:** Hello. I have an issue...

Categories

- htaccess (18)
- SEO (1)
- CSS (3)
- phpBB (2)
- Web (5)
- Javascript (2)
- WordPress (3)
- Dreamweaver (2)
- Security (15)

```
# OR THIS FOR SIMPLER
```

```
Header set P3P "policyref=\"/w3c/p3p.xml\""
```

Make any file be a certain filetype (regardless of name or extension)

```
#Makes image.gif, blah.html, index.cgi all act as php
```

```
ForceType application/x-httpd-php
```

Redirect non-https requests to https server fixing double-login problem and ensuring that htpasswd authorization can only be entered using HTTPS

Additional https/ssl information and Apache SSL in htaccess examples

```
SSLOptions +StrictRequire
```

```
SSLRequireSSL
```

```
SSLRequire %{HTTP_HOST} eq "google.com"
```

```
ErrorDocument 403 https://google.com
```

SEO Friendly redirects for bad/old links and moved links

For single moved file

```
Redirect 301 /d/file.html http://www.htaccesselite.com/r/file.html
```

For multiple files like a blog/this.php?gh

```
RedirectMatch 301 /blog(.*) http://www.askapache.com/$1
```

different domain name

```
Redirect 301 / http://www.newdomain.com
```

- DreamHost (2)
- ShellScript (1)
- Email (1)
- PHP (3)
- Unix (2)
- Windows (1)

Pages

- About
 - Comments Policy
 - Copyright Policy
 - Glossary
 - Legal Notice
 - Terms and Conditions
- Contact

Links

- **Apache Friends**
 - modules.apache.org
 - Apache Lounge
 - Official Apache Site
 - Apache Labs
 - Apache News
 - Planet Apache
 - Apache Week
 - Apache Wiki
- **Articles**
 - A List Apart
- **Blogroll**
 - htaccessElite
 - Check-these.info
 - htaccess Google Group

Require the www

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_URI} !^/robots\.txt$

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

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

Require the www without hardcoding

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_URI} !^/robots\.txt$ [NC]

RewriteCond %{HTTP_HOST} !^www\.[a-z]+\.[a-z]{2,6} [NC]

RewriteCond %{HTTP_HOST} ([a-z]+\.[a-z]{2,6})$ [NC]

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

Require no subdomain

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_URI} !^/robots\.txt$

RewriteCond %{HTTP_HOST} \.([a-z]+\.[a-z]{2,6})$ [NC]

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

Require no subdomain

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{HTTP_HOST} \.([^\.]+\.[^\.0-9]+)$

RewriteCond %{REQUEST_URI} !^/robots\.txt$ [NC]

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

- o maxdesign
- o ServerWatch
- o php.net
- o webstandards.org

Cool


- o Hype-Music!
- o WireShark

Web Hosting

- o DreamHost


65 listeners
BY FEEDBURNER

Search this blog:



» Blogs that link here

» View my profile



Archives

- January 2007 (15)
- December 2006 (12)
- November 2006 (20)

Amazon Reads

-

Redirect everyone to different site except 1 IP address (useful for web-development)

```
ErrorDocument 403 http://www.someothersite.com  
  
Order deny,allow  
  
Deny from all  
  
Allow from 24.33.65.6
```

CHMOD your files

chmod .htpasswd files 640

chmod .htaccess files 644

chmod php files 600

chmod files that you really dont want people to see as 400

NEVER chmod 777, if something requires write access use 766

Variable (mod_env) Magic

Set the Timezone of the server:

```
SetEnv TZ America/Indianapolis
```

Set the Server Administrator Email:

```
SetEnv SERVER_ADMIN webmaster@htaccesselite.com
```

Turn off the ServerSignature

```
ServerSignature Off
```

Add a “en-US” language tag and “text/html; UTF-8” headers without meta tags

Article: [Setting Charset in htaccess](#)

Article: [Using FilesMatch and Files in htaccess](#)

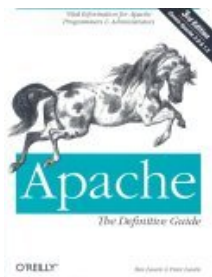
```
AddDefaultCharset UTF-8  
  
# Or AddType 'text/html; charset=UTF-8' html  
  
DefaultLanguage en-US
```

Using the Files Directive

```
<Files ~ "\.(htm|html|css|js|php)$">  
  
    AddDefaultCharset UTF-8  
  
    DefaultLanguage en-US  
  
</Files>
```

Using the FilesMatch Directive (*preferred*)

```
<FilesMatch "\.(htm|html|css|js|php)$">  
  
    AddDefaultCharset UTF-8  
  
    DefaultLanguage en-US  
  
</FilesMatch>
```



Use a custom php.ini with mod_php or php as a cgi

Article: [Custom PHP.ini tips and tricks](#)

When php run as Apache Module (mod_php)

in root .htaccess

```
SetEnv PHPRC /location/todir/containing/phpinifile
```

When php run as CGI

Place your php.ini file in the dir of your cgi'd php, in this case /cgi-bin/

htaccess might look something like this

```
AddHandler php-cgi .php .htm Action php-cgi /cgi-bin/php5.cgi
```

When cgi'd php is run with wrapper (for FastCGI)

You will have a shell wrapper script something like this:

```
#!/bin/sh

export PHP_FCGI_CHILDREN=3

exec /user3/x.com/htdocs/cgi-bin/php5.cgi
```

Change To

```
#!/bin/sh

export PHP_FCGI_CHILDREN=3

exec /x.com/cgi-bin/php.cgi -c /abs/path/to/php.ini
```

Securing directories: Remove the ability to execute scripts

Heres a couple different ways I do it

```
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi

Options -ExecCGI
```

This is cool, you are basically categorizing all those files that end in those extensions so that they fall under the jurisdiction of the -ExecCGI command, which also means -FollowSymLinks (*and the opposite is also true, +ExecCGI also turns on +FollowSymLinks*)

Only allow GET and PUT request methods to your server.

```
Options -ExecCGI -Indexes -All +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_METHOD} !^(GET|PUT)

RewriteRule .* - [F]
```

Processing All gif files to be processed through a cgi script

```
Action image/gif /cgi-bin/filter.cgi
```

Process request/file depending on the request method

```
Script PUT /cgi-bin/upload.cgi
```

Force Files to download, not be displayed in browser

```
AddType application/octet-stream .avi
AddType application/octet-stream .mpg
```

Then in your HTML you could just link directly to the file..

```
<a href="/movies/mov1.avi">Download Movie1</a>
```

And then you will get a pop-up box asking whether you want to save the file or open it.

Show the source code of dynamic files

If you'd rather have .pl, .py, or .cgi files displayed in the browser as source rather than be executed as scripts,

simply create a .htaccess file in the relevant directory with the following:


```
RemoveHandler cgi-script .pl .py .cgi
```

Dramatically Speed up your site by implementing Caching!

Article: [Speed Up Sites with htaccess Caching](#)

```
# MONTH
<FilesMatch "\.(flv|gif|jpg|jpeg|png|ico|swf)$">
    Header set Cache-Control "max-age=604800"
</FilesMatch>

# WEEK
<FilesMatch "\.(js|css|pdf|txt)$">
    Header set Cache-Control "max-age=604800"
</FilesMatch>

# DAY
<FilesMatch "\.(html|htm)$">
    Header set Cache-Control "max-age=60,must-revalidate"
</FilesMatch>
```

Prevent Files image/file hotlinking and bandwidth stealing

```
Options +FollowSymLinks

RewriteEngine On

RewriteBase /

RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?askapache.com/*$ [NC]
RewriteRule \.(gif|jpg|swf|flv|png)$ http://www.askapache.com/evil-hotlinker.gif [R=302,L]
```

ErrorDocuments

Article: [Additional ErrorDocument Info and Examples](#)

```
ErrorDocument 404 /favicon.ico
ErrorDocument 403 https://secure.htaccesselite.com
```

```
ErrorDocument 404 /cgi-bin/error.php
ErrorDocument 400 /cgi-bin/error.php
ErrorDocument 401 /cgi-bin/error.php
ErrorDocument 403 /cgi-bin/error.php
ErrorDocument 405 /cgi-bin/error.php
ErrorDocument 406 /cgi-bin/error.php
ErrorDocument 409 /cgi-bin/error.php
ErrorDocument 413 /cgi-bin/error.php
ErrorDocument 414 /cgi-bin/error.php
ErrorDocument 500 /cgi-bin/error.php
ErrorDocument 501 /cgi-bin/error.php
```

Note: You can also do an external link, but don't do an external link to your site or you will cause a loop that will hurt your SEO.

Authentication Magic

Require password for 1 file:

```
<Files login.php>
    AuthName "Prompt"
    AuthType Basic
    AuthUserFile /home/askapache.com/.htpasswd
    Require valid-user
</Files>
```

Protect multiple files:

```
<FilesMatch "^(exec|env|doit|phpinfo|w)\.*$">
    AuthName "Development"
    AuthUserFile /.htpasswd
    AuthType basic
    Require valid-user
</FilesMatch>
```

Example uses of the Allow Directive:

```
# A (partial) domain-name
Allow from 10.1.0.0/255.255.0.0

# Full IP address
Allow from 10.1.2.3

# More than 1 full IP address
Allow from 192.168.1.104 192.168.1.205

# Partial IP addresses
# first 1 to 3 bytes of IP, for subnet restriction.
Allow from 10.1
Allow from 10 172.20 192.168.2

# network/netmask pair
Allow from 10.1.0.0/255.255.0.0

# network/nnn CIDR specification
Allow from 10.1.0.0/16

# IPv6 addresses and subnets
Allow from 2001:db8::a00:20ff:fea7:ccea
Allow from 2001:db8::a00:20ff:fea7:ccea/10
```

Using visitor dependent environment variables:

Article: [Additional SetEnvIf examples](#)

```
SetEnvIf User-Agent ^KnockKnock/2\.0 let_me_in

Order Deny,Allow

Deny from all

Allow from env=let_me_in
```

Allow from apache.org but deny from foo.apache.org

```
Order Allow,Deny

Allow from apache.org

Deny from foo.apache.org
```

Allow from IP address with no password prompt, and also allow from non-Ip address with password prompt:

```
AuthUserFile /home/www/sites1-passwd
AuthType Basic
AuthName MySite
Require valid-user
Allow from 172.17.10
Satisfy Any
```

block access to files during certain hours of the day

```
Options +FollowSymLinks
RewriteEngine On
RewriteBase /
# If the hour is 16 (4 PM) Then deny all access
RewriteCond %{TIME_HOUR} ^16$
RewriteRule ^.*$ - [F,L]
```

A good default example .htaccess file

I use this when I start a new site, and uncomment or delete parts of the file depending on the sites needs

```
#           DEFAULT SETTINGS
#####
Options +ExecCGI -Indexes
DirectoryIndex index.php index.html index.htm

### DEFAULTS ###
ServerSignature Off
AddType video/x-flv .flv
AddType application/x-shockwave-flash .swf
AddType image/x-icon .ico
AddDefaultCharset UTF-8
DefaultLanguage en-US
SetEnv TZ America/Indianapolis
```

```
SetEnv SERVER_ADMIN webmaster@^^SITE^^.^^TLD^^

### FAST-CGI ###
AddHandler fastcgi-script fcgi
AddHandler php-cgi .php
Action php-cgi /cgi-bin/php5-wrapper.fcgi

#           HEADERS and CACHING
#####
#### CACHING ####
# WEEK
<FilesMatch ".(flv|gif|jpg|jpeg|png|ico)$">
    Header set Cache-Control "max-age=604800"
</FilesMatch>
# WEEK
<FilesMatch ".(js|css|pdf|swf)$">
    Header set Cache-Control "max-age=604800"
</FilesMatch>
# 10 minutes
<FilesMatch ".(html|htm|txt)$">
    Header set Cache-Control "max-age=600,must-revalidate"
</FilesMatch>
# DONT CACHE
<FilesMatch ".(pl|php|cgi|spl|scgi|fcgi)$">
    Header unset Cache-Control
</FilesMatch>

#           REWRITES AND REDIRECTS
#####
### SEO REDIRECTS ###
Redirect 301 /2006/uncategorized/htaccesslitecom-aboutus.html http://www.^^SITE^^.^^TLD^^

### REWRITES ###
RewriteEngine On
RewriteBase /

### WORDPRESS ###
# BEGIN WordPress
```

```
RewriteEngine On

RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]

</IfModule>

# END WordPress

# AUTHENTICATION
#####
AuthName "^^SITE^^.^^TLD^^"
Require valid-user
AuthUserFile /^^SITE^^.^^TLD^^/.htpasswd
AuthType basic
```

Stop browser prefetching

```
RewriteEngine On

SetEnvIfNoCase X-Forwarded-For .+ proxy=yes
SetEnvIfNoCase X-moz prefetch no_access=yes

# block pre-fetch requests with X-moz headers
RewriteCond %{ENV:no_access} yes
RewriteRule .* - [F,L]
```

Make a prefetching hint for Firefox.

```
Header append Link "</index.htm>; rel=prefetch"
```

Getting prefetching to show up in your logs

Getting prefetching to show up in our logs

First of all, how do we know a prefetch when we see one?

Firefox puts a header in each prefetching request, like this:

```
X-moz: prefetch
```

So we'll need to ask our web server to trap that information and log it somewhere useful. The options are:

Make a separate log file, just for prefetching requests.

Add an extra field to our log file format.

Mush something about prefetching into an existing field in our log file.

I have enough log files as it is, and I don't want to confuse my log analysis software by adding a custom field, so I'm going to squidge the X-Moz header onto the end of the User-Agent field of my current logs. (They're in "combined" format, which includes a field for the referer). Log analysis software will usually ignore crap tagged on the end of the User-Agent field, so this will tell me which hits have been prefetched without breaking anything else.

Let's tell Apache about the new log format we're inventing. We'll call the format

"combined_with_prefetching_hack".

Somewhere in our apache configuration file (httpd.conf or apache2.conf) we should have a line like this.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

Underneath that, we'll add another line like this:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i %{X-Moz}i\""   
combined_with_prefetching_hack
```

Then we'll find the place where we are currently telling apache to use the "combined" format for our site, and tell it to use "combined_with_prefetching_hack" instead.

Comment out a line a bit like this:

```
CustomLog /var/log/apache2/access.log combined
```

and replace it with something more like this:

```
CustomLog /var/log/apache2/access.log combined_with_prefetching_hack
```

then restart apache.

Now if we want our log file without the prefetched stuff: `grep -v prefetch access.log`

Related Entries

- [Allowing Access From 1 static IP and Deny the rest](#)
- [using last-modified, etag, mod_expires, mod_headers to seo phpBB](#)
- [Best htaccess tutorials and articles](#)
- [Set Environment Variables According To URL](#)
- [Ultimate htaccess File sample](#)

Viewed 42654 times | Dec 04, 06 | | Posted in [htaccess](#)

★★★★★★★ (8.86)

3 Responses to “Ultimate htaccess Article”

1. Max Design - standards based web design, development and training » Some links for on 05 Dec 2006 at 12:15

light reading (5/12/06) said ...

am

Featured on MaxDesign.com.au!

2. tenstar said ...

on 07 Dec 2006 at 1:40

am

That's perfect summary, very valuable for my next job of doing SEO friendly urls

through htaccess. Thank you.

3. webmaster said ...

on 09 Jan 2007 at 8:25

am

Round robin DNS is a technique in which load balancing is performed by a DNS server

instead of a strictly dedicated machine. This technique is usually only implemented on large networks or on IRC networks.

Round robin works by responding to DNS requests not with a single IP address, but a list of IP addresses (all of

which would assumedly host the same content). The order in which IP addresses from the list are returned is the basis of the round robin name. The IP address at the top of the list is returned a set number of times before it is moved to the bottom, thus promoting the second IP address to the top of the list. This cycle is continual and allows the DNS server to assist in balancing requests between servers.

Basic DNS clients simply use the first entry and discard the rest. However, there is no internet standard for deciding which address will be used by the requesting application - a few resolvers even re-order the list to give priority to numerically "closer" networks. Some desktop clients may even try alternate addresses after a connection time out of 30-45 seconds.

Round robin DNS is usually used for balancing the load of geographically-distributed Web servers. For example, a company has one domain name and three identical home pages residing on three servers with three different IP addresses. When one user accesses the home page it will be sent to the first IP address. The second user who accesses the home page will be sent to the next IP address, and the third user will be sent to the third IP address. In each case, once the IP address is given out, it goes to the end of the list. The fourth user, therefore, will be sent to the first IP address, and so forth.

Many IRC networks use round robin DNS to distribute users across the servers on their networks. Indeed, virtually all the large and established networks have separate round robin DNS setup for each continent or country in which they have servers - so users can use a 'random' server local to them.

Although very easy to implement, round robin DNS has important drawbacks, such as those inherited from the DNS hierarchy itself and TTL (Time to live) values, which allows for address caching and can be very difficult to manage. Round robin DNS must not solely be relied upon for service availability. If a service at one of the addresses in the list goes down, the DNS will continue to hand out that address and clients will still attempt to reach the dead service.

Also, it may not be the best choice for load balancing on its own since it merely alternates the order of the address records each time a nameserver is queried. There is no consideration for transaction time, server load, network congestion, etc. Round robin DNS load balancing will only work for services with a large number of

uniformly distributed connections to servers of equivalent capacity. Otherwise it just does load distribution.

To overcome this limitation, techniques can be used to combine 'traditional' DNS round robin techniques with scripts that routinely poll servers mirroring content to see if they are online, and not overloaded, etc. If a server does not return a reply as expected, the server can be temporarily removed from the DNS pool, until it reports that it is once again operating normally. By geographically distributing among disparate hosts, such a scripted-solution can be an ultra-cheap and reliable work-around, towards maintaining a nearly 24/7 reliable global server up-time.

[Trackback This Post](#) | [Subscribe to the comments through RSS Feed](#)

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website