

```
# MODX supports Friendly URLs via this .htaccess file. You must serve web
# pages via Apache with mod_rewrite to use this functionality, and you must
# change the file name from ht.access to .htaccess.
#
# Make sure RewriteBase points to the directory where you installed MODX.
# E.g., "/modx" if your installation is in a "modx" subdirectory.
#
# You may choose to make your URLs non-case-sensitive by adding a NC directive
# to your rule: RewriteRule ^(.*)$ index.php?q=$1 [L,QSA,NC]
```

```
RewriteEngine On
```

```
RewriteBase /
```

```
# Rewrite www.domain.com -> domain.com -- used with SEO Strict URLs plugin
```

```
RewriteCond %{HTTP_HOST} .
```

```
RewriteCond %{HTTP_HOST} !^supremeautosupplies\.co\.uk [NC]
```

```
RewriteRule (.*) http://supremeautosupplies.co.uk/$1 [R=301,L]
```

```
#
```

```
# or for the opposite domain.com -> www.domain.com use the following
```

```
# DO NOT USE BOTH
```

```
#
```

```
#RewriteCond %{HTTP_HOST} .
```

```
#RewriteCond %{HTTP_HOST} !^www\.example-domain-please-change\.com [NC]
```

```
#RewriteRule (.*) http://www.example-domain-please-change.com/$1 [R=301,L]
```

```
# Rewrite secure requests properly to prevent SSL cert warnings, e.g. prevent
```

```
# https://www.domain.com when your cert only allows https://secure.domain.com
```

```
#RewriteCond %{SERVER_PORT} !^443
```

```
#RewriteRule (.*) https://example-domain-please-change.com/$1 [R=301,L]
```

```
# The Friendly URLs part
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
```

```
# Make sure .htc files are served with the proper MIME type, which is critical
```

```
# for XP SP2. Un-comment if your host allows htaccess MIME type overrides.
```

```
#AddType text/x-component .htc
```

```
# If your server is not already configured as such, the following directive
```

*# If your server is not already configured as such, the following directive  
# should be uncommented in order to set PHP's register\_globals option to OFF.  
# This closes a major security hole that is abused by most XSS (cross-site  
# scripting) attacks. For more information: [http://php.net/register\\_globals](http://php.net/register_globals)  
#*

*# To verify that this option has been set to OFF, open the Manager and choose  
# Reports -> System Info and then click the phpinfo() link. Do a Find on Page  
# for "register\_globals". The Local Value should be OFF. If the Master Value  
# is OFF then you do not need this directive here.*

*#  
# IF REGISTER\_GLOBALS DIRECTIVE CAUSES 500 INTERNAL SERVER ERRORS :*

*# Your server does not allow PHP directives to be set via .htaccess. In that  
# case you must make this change in your php.ini file instead. If you are  
# using a commercial web host, contact the administrators for assistance in  
# doing this. Not all servers allow local php.ini files, and they should  
# include all PHP configurations (not just this one), or you will effectively  
# reset everything to PHP defaults. Consult [www.php.net](http://www.php.net) for more detailed  
# information about setting PHP directives.*

*#php\_flag register\_globals Off*

*# For servers that support output compression, you should pick up a bit of  
# speed by un-commenting the following lines.*

*#php\_flag zlib.output\_compression On  
#php\_value zlib.output\_compression\_level 5*

*# The following directives stop screen flicker in IE on CSS rollovers. If  
# needed, un-comment the following rules. When they're in place, you may have  
# to do a force-refresh in order to see changes in your designs.*

*#ExpiresActive On  
#ExpiresByType image/gif A2592000  
#ExpiresByType image/jpeg A2592000  
#ExpiresByType image/png A2592000  
#BrowserMatch "MSIE" brokenvary=1  
#BrowserMatch "Mozilla/4.[0-9]{2}" brokenvary=1  
#BrowserMatch "Opera" !brokenvary  
#SetEnvIf brokenvary 1 force-no-vary*