

Emezeta **mod_rewrite** Card Reference Cheat Sheet

version 0.1

Apache mod_rewrite directives, options and parameters by Manz

<http://www.emezeta.com>

DirectiveName description and info

directive syntax <parameter> [optional-parameter]



```
Example code line 1
Example code line 2
Example code line 3
```

original request => final rewrite

Note, warnings and information

Legend

RewriteEngine enable/disable rewriting engine

RewriteEngine <on|off>

RewriteBase set base URL for rewrites

RewriteBase <dir>



```
RewriteEngine On
RewriteBase /dir
RewriteRule ^oldfile\.html$ newfile.html
```

/oldfile.html => /dir/newfile.html

RewriteMap define map func for key-lookup

RewriteMap <map-name> <map-type:map-source>

map-name Map function reference name

map-type:map-source

txt: standard plain text :path .txt regular file

rnd: random option plain text :path .rnd regular file

dbm[=type]: hash value :path .db regular file [Apache 2.0.41+]

int: internal function :func apache function

:toupper convert to all uppercase

:tolower convert to all lowercase

:escape translate to hex-coding

:unescape translate to special chars

prg: external program :path program regular file



```
RewriteEngine On
RewriteMap funcname txt:/path/map.txt
RewriteRule ^/dir/(.*)$ ${funcname:$1|default}
```

Note: do NOT use 'RewriteMap' in .htaccess, use it in apache main conf file!

txt-file format

Alfred.J.Quack
Jonathan.Jack.Oneill11

ajq
jjo

rnd-file format

option1 www1|www2|www3
option2 www5|www6

RewriteLock set lock filename for RewriteMap sync

RewriteLock <file-path>



```
RewriteEngine On
RewriteMap translate prg:/path/trans.pl
RewriteRule ^oldfile\.html$ newfile.html
RewriteLock /home/user/rewrite/lockfile
```

RewriteOptions set options for rewrite engine

RewriteOptions <Options>

inherit force to inherit parent's configuration

MaxRedirects=N limit to N redirects per request. If more,
give a HTTP error 500. (default: 10 redirect limit) [Apache 2.0.45+]



```
RewriteEngine On
RewriteOptions MaxRedirects=25
RewriteRule ^oldfile\.html$ newfile.html
```

RewriteLog set filename for logging rewrite processing

RewriteLog <file-path>

RewriteLogLevel set verbosity level of the logging rewriting

RewriteLogLevel <Level>

level 0: not logging (default), 1:, 9: all process logging



```
RewriteEngine On
RewriteLog "/var/log/modrewrite_log"
RewriteLogLevel 3
```

Note: Use log level > 2 only for debugging. Too much activity!

Note: Do not rewrite log in /dev/null. To disable use RewriteLogLevel 0.

RewriteCond define condition under which rewriting

RewriteCond <Test-string> <Cond-pattern> [[flags]]

Test-string target variable or information

\$N (1...9) Reference to grouped parts in RewriteRule (parentheses)

%N (1...9) Reference to grouped parts in RewriteCond (parentheses)

RewriteMap Reference to RewriteMap (expansion map function)

Server Vars Reference to %{SERVER-VARIABLES}

HTTP_USER_AGENT

REMOTE_ADDR

API_VERSION

HTTP_REFERER

REMOTE_HOST

IS_SUBREQ

HTTP_COOKIE

REQUEST_METHOD

DOCUMENT_ROOT

HTTP_FORWARDED

QUERY_STRING

SERVER_NAME

HTTP_HOST

HTTP_ACCEPT

REQUEST_URI

%{ENV:var} Reference to environment Apache vars (getenv)

%{SSL:var} Reference to SSL environment vars (mod_ssl)

%{HTTP:var} Reference to HTTP header vars (MIME header)

%{LA-U:var} Reference to internal URL-based sub-request vars

%{LA-F:var} Reference to internal file-based sub-request vars

Cond-pattern regexp which is applied to previous Test-string

!pattern non-matching pattern

<pattern true if test-string lexicographically precedes pattern

>pattern true if test-string lexicographically follows pattern

=pattern true if test-string are exactly equal (char for char)

-d exist and is directory

-l exist and is a symbolic link

-f exist and is a regular file

-F exist and is a valid file request

-s same -f and > 0 bytes

-U exist and is a valid URL request

Flags

NC|nocase case-insensitive ('A' is equal to 'a')

OR|ornext combine rule with next rule as OR operator



```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^.*MSIE.*$ [OR]
RewriteCond %{HTTP_USER_AGENT} ^Mozilla/3.*$
RewriteRule ^/$ old_compliant.html
```

Note: Several RewriteCond implicit AND operator.

RewriteRule define rules for the rewriting engine

RewriteRule <Pattern> <Substitution> [[flags]]

Pattern regexp applied to current URL

. any single character

[chars] char-class: any char of the class

[^chars] char-class: not a char of the class

text1|text2 alternative: text1 or text2

?

0 or 1 occurrences of previous char

* 0 or more occurrences of previous char

+ 1 or more occurrences of previous char

(text)

grouping of text (use to create \$N or %N)

^

start-of-line anchor

\$

end-of-line anchor

\char

literally char (special chars' escape)

Substitution string substituted or replace for pattern matched

Flags

C|chain chain current rule with next rule (only if all matched)

CO=N:V:D[:L[:P]]|cookie set cookie (N: name, V: value,
D: domain, L: lifetime, P: path)

E=V:VV|env set environment var (V: var-name, VV: value)

F|forbidden force current URL to be forbidden (403 error)

G|gone force current URL to be gone (410 error)

L|last stop (break) rewriting and don't apply any more rule

N|next skip (continue) next rules and start again first rule

NC|nocase case-insensitive ('A' is equal to 'a')

NE|noescape not apply URI hexcode escape ('%', '\$', ';', ...)

NS|nosubreq skip all internal sub-requests

P|proxy force proxy (use mod_proxy module)

PT|passthrough pass through to next handler

QSA|qsappend append query string part instead of replace it

R=N|redirect force a redirection with N code (default: 302)

S=N|skip skip next N rules (not same chain flag!)

T=MIME|type force MIME-type of the target file



```
RewriteEngine On
RewriteRule ^/zone1/?$ /zone/ [R=301,L]

RewriteRule ^/robots.txt$ /robots.txt.php [S=4,NC]
```

Note: Several RewriteCond implicit AND operator.