

RewriteCond

The `*RewriteCond*` directive defines conditions under which the rewriting should take place. h2. Syntax We should really encourage people to use the lexicographically equal operator instead of a [RegEx] if they want to ckeck, if `_test string_` is lexicographically equal to `_cond pattern_`. E.g. using `{noformat} RewriteCond %{HTTP_HOST} !="" {noformat}` instead of `{noformat} RewriteCond %{HTTP_HOST} . # or RewriteCond %{HTTP_HOST} !^$ {noformat}` or using `{{{RewriteCond \{REQUEST_URI\} !=/foo/bar }}} {noformat}` instead of `{noformat} RewriteCond %{REQUEST_URI} !=/foo/bar$ {noformat}` or `{{{RewriteCond \{SERVER_PORT\} =443 }}} {noformat}` instead of `{noformat} RewriteCond %{SERVER_PORT} ^443$ {noformat}` *Note:* Conditions are being processed after the pattern of the [RewriteRule] has matched. This means that the Condition in following example would be useless (it's always true): `{noformat} RewriteCond %{REQUEST_URI} \.html$ RewriteRule \.html$ - [G] {noformat}` While this one wastes performance: `{noformat} RewriteCond %{REQUEST_URI} ^/([^\.]*)\.html$ RewriteRule ^/(.*) /%1.php [PT] {noformat}` Every Request matches the rule-pattern and after that the condition will be checked. But you can easily check the uri value in the rule-pattern, so that there is no need for such a condition here: `{noformat} RewriteRule ^/([^\.]*)\.html$ /$1.php [PT] {noformat}` h2. Examples See [ConditionalRewrites]