mod_perl is an Apache module, created by Doug MacEachern, that embeds a Perl interpreter within the server. It provides a Perl API to Apache and adds a number of Apache configuration directives. Scripts using mod_perl should import the Apache module, Apache::Constants, and other Apache:: modules. A reference to the request object (denoted below by $r) is passed to Perl handlers when they are invoked.

Client request methods
- $str = $r->uri();
- $str = $r->the_request();
- $str = $r->connection();
- $str = $r->content_encoding();
- $str = $r->filename();
- $str = $r->get_remote_logname();
- $str = $r->get_remote_host();
- $str = $r->header(in($hdr, $newval));
- $bool = $r->header_only();
- $str = $r->headers_in();
- $str = $r->headers_out();
- $str = $r->internal_redirect_handler($newplace);
- $bool = $r->is_main();
- $str = $r->last();
- $req = $r->next();
- $req = $r->prev();
- $log = $r->log();
- $bool = $r->main();
- $bool = $r->notes($k);
- $bool = $r->notes();
- $log = $r->print(@list);
- $log = $r->printf($format, @args);
- $r->rflush();
- $r->cgi_header_out($hdr);
- $r->content_languages($aref);
- $str = $r->content_encoding();
- $r->register_cleanup($code_ref);
- $tab = $r->notes();
- # use :methods tag
- $tab->add($key, $str_or_aref);
- $tab->clear();
- $tab->do($code_ref);
- $tab->set($key, $str);
- $tab->unset($key);
- $str = $r->request();
- $str = $r->uri($newval);

Server response methods
- $num = $r->bytes_sent();
- $str = $r->cgi_header_out($hrdr, $newval);
- $str = $r->content_encoding($hrdr, $newval);
- $newval = $r->handler($hrdr, $newval);
- $str = $r->header_out($hrdr, $newval);
- $newval = $r->headers_out($hrdr, $newval);
- $newval = $r->headers_out();
- $bool = $r->next_time($hrdr, $newval);
- $bool = $r->status($hrdr, $newval);
- $str = $r->status_line($hrdr, $newval);

Sending data to the client
- $str = $r->args();
- $str = $r->begin_request();
- $r->send_http_header([$content_type]);
- $r->send_fd($fd, $filehandle);
Apache::Constants class

The following export tag groups are defined (HTTP status code synonyms are given in brackets):

```perl
:common: OK, DECLINED, DONE, NOT FOUND, FORBIDDEN, AUTH_REQUIRED (HTTP_UNAUTHORIZED), SERVER_ERROR
:response: DOCUMENT_FOLLOWS (HTTP_OK), MOVE (HTTP_MOVED_PERMANENTLY), REDIRECT (HTTP_MOVED_TEMPORARILY), USE_LOCAL_COPY (HTTP_NOT_MODIFIED), BAD_REQUEST, BAD_GATEWAY, NOT_IMPLEMENTED, CONTINUE, NOT_AUTHORITATIVE
:methods: CONNECT, COPY, DELETE, GET, HEAD, IF_MATCH, IF_MODIFIED_SINCE, OPTIONS, PATCH, POST, PROPFIND, PROPFIND, PUT, TRACE, UNLOCK, METHODS
:options: OPT_ALL, OPT_NONE, OPT_INDEXES, OPT_INCLUDES, OPT_SYNC_LINKS, OPT_EXECDGI, OPT_UNSET, OPT_INCHNEXEC, OPT_SYM_OWNER, OPT_MULTI.
:satisfies: SATISFIES_ALL, SATISFY_ANY, SATISFY_NOSPEC
:server: MODULE_MAGIC_NUMBER, SERVER_BUILD, SERVER_VERSION
:remotehost: REMOTE_HOST, REMOTE_NAME, REMOTE_NOLOOKUP, REMOTE_DOUBLE_REV
```

304 HTTP_SEE_OTHER

```perl
:HTTP_OK
:HTTP_CREATED
:HTTP_ACCEPTED
:HTTP_ACCEPT
:HTTP_NO_CONTENT
:HTTP_RESET_CONTENT
:HTTP_PARTIAL_CONTENT
:HTTP_MULTIPLE_CHOICES
:HTTP_MOVED_PERMANENTLY
:HTTP_MOVED_TEMPORARILY
:HTTP_SEE_OTHER
:HTTP_NOT_MODIFIED
:HTTP_USE_PROXY
:HTTP_BAD_REQUEST
:HTTP_UNAUTHORIZED
:HTTP_FORBIDDEN
:HTTP_NOT_FOUND
```

404 HTTP_NOT_FOUND

```
:Magic global variables
$6, $5, $8, $9, $10, $11, $12, $13, $14, $15
```

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept: media-types[; q=value] [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Accept-Charset: charset[; q=value] [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Accept-Encoding: encoding[; q=value] [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Accept-Language: lang[; q=value] [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Accept-Ranges: [bytes][; none]</td>
<td>RESPONSE</td>
</tr>
<tr>
<td>Age: seconds</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Allow: method [; ...]</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Authorization: scheme credentials</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Cache-Control: directive</td>
<td>GENERAL</td>
</tr>
<tr>
<td>Connection: close</td>
<td>GENERAL</td>
</tr>
<tr>
<td>Content-Base: uri</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-Encoding: enc</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-Language: lang</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-Length: len</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-MD5: digest</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-Range: bytes range/length</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Content-Type: media-type</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Cookie: name=value [; ...]</td>
<td>GENERAL</td>
</tr>
<tr>
<td>Date: date</td>
<td>RESPONSE</td>
</tr>
<tr>
<td>ETag: entity-tag</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Expect: expectation</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Expires: date</td>
<td>REQUEST</td>
</tr>
<tr>
<td>From: email-address</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Host: hostname[; port]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>If-Match: entity-tag</td>
<td>REQUEST</td>
</tr>
<tr>
<td>If-Modified-Since: date</td>
<td>REQUEST</td>
</tr>
<tr>
<td>If-None-Match: entity-tag</td>
<td>REQUEST</td>
</tr>
<tr>
<td>If-Range: [entity tag][date]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>If-Unmodified-Since: date</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Last-Modified: date</td>
<td>ENTITY</td>
</tr>
<tr>
<td>Location: uri</td>
<td>RESPONSE</td>
</tr>
<tr>
<td>MIME-Version: version</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Max-Fowrards: number</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Pragma: [no-cache][;extension-pragma]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Proxy-Authenticate: challenge</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Proxy-Authorization: credentials</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Public: method</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Range: bytes[; -][; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Referer: url</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Retry-After: [date][seconds]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Server: string</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Set-Cookie: name=value[; options]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>TE: coding</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Trailer: header</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Transfer-Encoding: coding</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Upgrade: protocol [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>User-Agent: string</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Vary: header [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Via: [;protocol][;version] [;comment] [; ...]</td>
<td>REQUEST</td>
</tr>
<tr>
<td>WWW-Authenticate: scheme realm</td>
<td>REQUEST</td>
</tr>
<tr>
<td>Warning: code agent <em>text</em> [date]</td>
<td>REQUEST</td>
</tr>
</tbody>
</table>

HTTP 1.1 headers

405 HTTP_METHOD_NOT_ALLOWED

```
:Quick reference cards
http://www.modperl.com
http://www.perl.com
```

Apache mod_perl configuration directives

mod_perl enables Apache to be configured using Perl statements that are contained within `<Perl>...` sections and adds the Apache configuration directives listed below. Each directive is given with its arguments; defaults are given where appropriate in parentheses at the end of the line, followed by the symbol `❖` to mark directives only valid in a directory section or .htaccess file:

- PerlAccessHandler handler
- PerlAuthenHandler handler
- PerlAuthzHandler handler
- PerlChildExitHandler handler
- PerlChildInitHandler handler
- PerlCleanupHandler handler
- PerlDispatchHandler handler
- PerlFixupHandler handler
- PerlFreshRestart [On|Off] (On)
- PerlHandler handler
- PerlHeaderParserHandler handler
- PerlInitHandler handler
- PerlLogHandler handler
- PerlModule
- PerlPassEnv name ...
- PerlPostReadRequestHandler handler
- PerlRequire script-file
- PerlSendHeader [On|Off] (Off)
- PerlSetEnv name value
- PerlSetVax name value
- PerlSetupEnv [On|Off] (Off)
- PerlTaintCheck [On|Off] (Off)
- PerlTransHandler handler
- PerlTypeHandler handler
- PerlWarn [On|Off] (Off)

Resources

- http://perl.apache.org
- Writing Apache Modules home page
- http://www.modperl.com
- Apache mod_perl
- http://www.perl.com
- http://www.refcards.com

Quick reference cards

```
mod_perl Quick Reference Card
A refcards.com™ quick reference card
Revision 1.0 for mod_perl version 1.19 [May 1999]
© 1998, 1999 Ford & Mason Ltd. All rights reserved.
Permission is granted to print and duplicate this card for personal or individual, internal business use. Copies of this card (& others) can be ordered through our web site: http://www.refcards.com, which also has versions available for downloading.
Please send feedback to: feedback@refcards.com
refcards.com is a trademark of Ford & Mason Ltd.
Use of the Camel for Perl is a trademark of O’Reilly & Associates – used by permission.
```