



## mod\_perl Quick Reference Card

Revision 1.0 for mod\_perl version 1.19  
Andrew Ford [refcards.com](http://refcards.com)<sup>TM</sup>

**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

```
$r = Apache->request();
$str = $r->args();           # or %hash = ...
$c = $r->connection         # see Apache::Connection
$str = $r->content();        # or %hash = ...
$str = $r->filename( [$newval] );
    $r->finfo();
$str = $r->get_remote_host( [$lookup_type] );
    # use Apache::Constants::remotehost tag
$str = $r->get_remote_logname();
$str = $r->header_in( $hdr [, $newval] );
$bool = $r->header_only();
$href = $r->headers_in();    # or %hash = ...
$str = $r->method( [$newval] );
$num = $r->method_number( [$nv] ); # use :methods tag
$u = $r->parsed_uri();       # see Apache::URI
$str = $r->path_info( [$newval] );
$str = $r->protocol();
$bool = $r->proxyreq( [$newval] );
    $r->read( $buf, $bytes_to_read );
$s = $r->server              # see Apache::Server
$str = $r->the_request();
$str = $r->uri( [$newval] );
```

### Server response methods

```
$num = $r->bytes_sent();
    $r->cgi_header_out( $hdr [, $newval] );
$str = $r->content_encoding( [$newval] );
$aref = $r->content_languages( [$newval] );
$str = $r->content_type( [$newval] );
    $r->custom_response( $code, $uri );
$str = $r->err_header_out( $hdr [, $newval] );
$href = $r->err_headers_out(); # or %hash = ...
$str = $r->handler( [$newval] );
$str = $r->header_out( $hdr [, $newval] );
$href = $r->headers_out();    # or %hash = ...
$bool = $r->no_cache( [$newval] );
$num = $r->request_time();
$num = $r->status( [$newval] );
$str = $r->status_line( [$newval] );
```

### Sending data to the client

```
$r->print( @list );          # checks $|
$r->printf( $format, @args );
$r->rflush();
$r->send_cgi_header( $str );
$len = $r->send_fd( $filehandle );
```

```
$r->send_http_header( [$content_type] );
```

### Server core functions

```
$r->chdir_file( $file );
$r->child_terminate();
$r->hard_timeout( $msg );
$r->internal_redirect( $newplace );
$r->internal_redirect_handler( $newplace );
$bool = $r->is_initial_req();
$bool = $r->is_main();
    $r->kill_timeout();
$str = $r->location();
$req = $r->last();
$req = $r->main();
$req = $r->next();
$str = $r->notes( $k [, $v] ); # or $tab = $r->notes()
$req = $r->prev();
    $r->register_cleanup( $code_ref );
    $r->reset_timeout();
    $r->soft_timeout( $msg );
$str = $r->subprocess_env( [ $k [, $v] ] );
```

### Server configuration methods

```
$str = $r->dir_config( $k ); #or $tab = $r->dir_config()
$str = $r->document_root();
$str = $r->get_server_name();
$num = $r->get_server_port();
$str = $r->server_root_relative( [$obj] );
```

### Logging and the Apache::Log class

```
$str = $r->as_string();
    $r->log_reason( $message, $file );
    $r->log_error( $message );
    $r->warn( $message );
$log = $r->log();
$log = $s->log();
    $log->emerg ( { $str ... | $code_ref } );
    $log->alert ( { $msg ... | $code_ref } );
    $log->crit ( { $msg ... | $code_ref } );
    $log->error ( { $msg ... | $code_ref } );
    $log->warn ( { $msg ... | $code_ref } );
    $log->notice( { $msg ... | $code_ref } );
    $log->info ( { $msg ... | $code_ref } );
    $log->debug ( { $msg ... | $code_ref } );
```

### Access control methods

```
$opts = $r->allow_options(); # use :options tag
$str = $r->auth_name( [$newval] );
$str = $r->auth_type();
($rc, $pw) = $r->get_basic_auth_pw();
    $r->note_basic_auth_failure();
$aref = $r->requires();
$flag = $r->satisfies();    # use :satisfies tag
$bool = $r->some_auth_required();
```

### mod\_perl specific methods

```
$str = $r->current_callback();
$bool = $r->define( $name );
    Apache->exit( [$code] );
$fh = Apache->gensym();
```

```
$aref = $r->get_handlers( $str );
    Apache->httpd_conf( $str );
$bool = $r->module( $name );
$bool = Apache->perl_hook( $name );
    $r->post_connection( $code_ref );
    $r->push_handlers( $str => $code_ref );
$r = Apache->request( [$r] );
    $r->set_handlers( $str => $aref );
```

### Apache::SubRequest class

```
$subr = $r->lookup_uri( $uri );
$subr = $r->lookup_file( $filename );
$src = $subr->run();
```

### Apache::Server class

```
$s = Apache->server          # or $r->server
$bool = $s->is_virtual();
    $s->log_error();
$aref = $s->names();
$s = $s->next();
$num = $s->port();
$str = $s->server_admin();
$str = $s->server_hostname();
$num = $s->timeout( [$newval] );
    $s->warn();
```

### Apache::Connection class

```
$bool = $c->aborted();
$str = $c->auth_type();
$addr = $c->local_addr();
$addr = $c->remote_addr( [$addr] );
$str = $c->remote_host();
$str = $c->remote_ip( [$ip] );
$str = $c->remote_logname();
$str = $c->user( [$username] );
```

### Apache::Table class

```
$tab = Apache::Table->new( $r [, $size] );
    $tab->add( $key, $str_or_aref );
    $tab->clear();
    $tab->do( $code_ref );
    $tab->merge( $key, $str_or_aref );
    $tab->set( $key, $str );
$str = $tab->get( $key );    # or @list = ...
    $tab->unset( $key );
```

### Apache::URI class

```
$uri = Apache::URI->parse( $r [, $string_uri] );
$str = $uri->unparse();
$str = $uri->component( [$newval] );
    (where component is one of: fragment, hostinfo, hostname,
    password, path_info, path, port, query, rpath, scheme, user)
```

### Apache::Util class

```
$str = Apache::Util::escape_html( $html );
$str = Apache::Util::escape_uri( $uri );
$str = Apache::Util::ht_time( $time [, $fmt [, $bool]] );
$secs = Apache::Util::parsedate( $date_str );
$num = Apache::Util::size_string( $num );
$str = Apache::Util::unescape_uri( $uri );
$str = Apache::Util::unescape_uri_info( $uri );
```

Apache::Constants class

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

```
:common: OK, DECLINED, DONE, NOT_FOUND, FORBIDDEN,
AUTH_REQUIRED (HTTP_UNAUTHORIZED), SERVER_ERROR

:response: DOCUMENT_FOLLOWS (HTTP_OK),
MOVED (HTTP_MOVED_PERMANENTLY),
REDIRECT (HTTP_MOVED_TEMPORARILY),
USE_LOCAL_COPY (HTTP_NOT_MODIFIED),
BAD_REQUEST, BAD_GATEWAY, NOT_IMPLEMENTED,
CONTINUE, NOT_AUTHORITATIVE

:methods: M_CONNECT, M_COPY, M_DELETE, M_GET, M_INVALID,
M_LOCK, M_MKCOL, M_MOVE, M_OPTIONS, M_PATCH, M_POST,
M_PROPFIND, M_PROPPATCH, M_PUT, M_TRACE, M_UNLOCK, METHODS

:options: OPT_ALL OPT_NONE, OPT_INDEXES, OPT_INCLUDES,
OPT_SYM_LINKS, OPT_EXECCGI, OPT_UNSET,
OPT_INCNOREXEC, OPT_SYM_OWNER, OPT_MULTII,

:satisfies: SATISFY_ALL, SATISFY_ANY, SATISFY_NOSPEC

:server: MODULE_MAGIC_NUMBER, SERVER_BUILT, SERVER_VERSION

:remotehost: REMOTE_HOST, REMOTE_NAME, REMOTE_NOLOOKUP,
REMOTE_DOUBLE_REV

:http includes only those HTTP status code constants shown below in
bold type (other HTTP constants may be imported explicitly):
100 HTTP_CONTINUE          405 HTTP_METHOD_NOT_ALLOWED
101 HTTP_SWITCHING_PROTOCOLS 406 HTTP_NOT_ACCEPTABLE
200 HTTP_OK                407 HTTP_PROXY_AUTHENTICATION_
                             REQUIRED
201 HTTP_CREATED
202 HTTP_ACCEPTED          408 HTTP_REQUEST_TIMEOUT
203 HTTP_NON_AUTHORITATIVE 409 HTTP_CONFLICT
204 HTTP_NO_CONTENT        410 HTTP_GONE
205 HTTP_RESET_CONTENT     411 HTTP_LENGTH_REQUIRED
206 HTTP_PARTIAL_CONTENT   412 HTTP_PRECONDITION_FAILED
300 HTTP_MULTIPLE_CHOICES  413 HTTP_REQUEST_ENTITY_TOO_LARGE
301 HTTP_MOVED_PERMANENTLY 414 HTTP_REQUEST_URI_TOO_LARGE
302 HTTP_MOVED_TEMPORARILY 415 HTTP_UNSUPPORTED_MEDIA_TYPE
303 HTTP_SEE_OTHER         500 HTTP_INTERNAL_SERVER_ERROR
304 HTTP_NOT_MODIFIED      501 HTTP_NOT_IMPLEMENTED
305 HTTP_USE_PROXY         502 HTTP_BAD_GATEWAY
400 HTTP_BAD_REQUEST       503 HTTP_SERVICE_UNAVAILABLE
401 HTTP_UNAUTHORIZED      504 HTTP_GATEWAY_TIME_OUT
402 HTTP_PAYMENT_REQUIRED  505 HTTP_VERSION_NOT_SUPPORTED
403 HTTP_FORBIDDEN         506 HTTP_VARIANT_ALSO_VARIES
404 HTTP_NOT_FOUND
```

Magic global variables

```
$0, $*X, $|, $/, %@, %SIG, @INC, %INC, %ENV{MOD_PERL},
%ENV{GATEWAY_INTERFACE}, %ENV{PERL_SEND_HEADER}
```

Special package globals

```
$Apache::Server::CWD          $Apache::Server::SaveConfig
$Apache::Server::ReStarting   $Apache::Server::Starting
```

HTTP 1.1 headers

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

Apache mod\_perl configuration directives

mod\_perl enables Apache to be configured using Perl statements that are contained within <Perl>...</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 <i>handler</i>	
PerlAuthenHandler <i>handler</i>	
PerlAuthzHandler <i>handler</i>	
PerlChildExitHandler <i>handler</i>	❖
PerlChildInitHandler <i>handler</i>	❖
PerlCleanupHandler <i>handler</i>	
PerlDispatchHandler <i>handler</i>	
PerlFixupHandler <i>handler</i>	
PerlFreshRestart {On Off}	(On) ❖
PerlHandler <i>handler</i>	
PerlHeaderParserHandler <i>handler</i>	
PerlInitHandler <i>handler</i>	
PerlLogHandler <i>handler</i>	
PerlModule	
PerlPassEnv <i>name</i> ...	❖
PerlPostReadRequestHandler <i>handler</i>	❖
PerlRequire <i>script-file</i>	
PerlSendHeader {On Off}	(Off)
PerlSetEnv <i>name value</i>	
PerlSetVar <i>name value</i>	
PerlSetupEnv {On Off}	(Off)
PerlTaintCheck {On Off}	(Off) ❖
PerlTransHandler <i>handler</i>	❖
PerlTypeHandler <i>handler</i>	
PerlWarn {On Off}	(Off) ❖

Resources

<a href="http://perl.apache.org">http://perl.apache.org</a>	The Apache/Perl Integration Project
<a href="http://www.modperl.com">http://www.modperl.com</a>	Writing Apache Modules home page
<a href="http://www.apache.org">http://www.apache.org</a>	Apache home page
<a href="http://www.perl.com">http://www.perl.com</a>	Perl home page
<a href="http://www.refcards.com">http://www.refcards.com</a>	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](mailto: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.