

1 Apache::RequestIO - Perl API for Apache request record IO

1.1 Synopsis

```
use Apache::RequestIO ();
```

META: to be completed

1.2 Description

`Apache::RequestIO` provides the API to perform IO on the Apache request object.

1.3 API

`Apache::RequestIO` provides the following functions and/or methods:

1.3.1 discard_request_body

META: Autogenerated - needs to be reviewed/completed

In HTTP/1.1, any method can have a body. However, most GET handlers wouldn't know what to do with a request body if they received one. This helper routine tests for and reads any message body in the request, simply discarding whatever it receives. We need to do this because failing to read the request body would cause it to be interpreted as the next request on a persistent connection.

```
$ret = $r->discard_request_body();
```

- **arg1: \$r** (`Apache::RequestRec`)

The current request

- **ret: \$ret** (integer)

error status if request is malformed, `Apache::OK` otherwise

1.3.2 setup_client_block

META: Autogenerated - needs to be reviewed/completed

META: I think this method is deprecated along with other `client_block` methods, use plain `$r-<read()` instead.

Setup the client to allow Apache to read the request body.

```
$ret = $r->setup_client_block($read_policy);
```

- **arg1: \$r** (`Apache::RequestRec`)

The current request

- **arg2: \$read_policy (Apache::RequestRec)**

How the server should interpret a chunked transfer-encoding. One of:

REQUEST_NO_BODY	Send 413 error if message has any body
REQUEST_CHUNKED_ERROR	Send 411 error if body without Content-Length
REQUEST_CHUNKED_DECHUNK	If chunked, remove the chunks for me.

- **ret: \$ret (integer)**

either OK or an error code

1.3.3 should_client_block

META: Autogenerated - needs to be reviewed/completed

META: I think this method is deprecated along with other client_block methods, use plain \$r-<read() instead.

Determine if the client has sent any data. This also sends a 100 Continue response to HTTP/1.1 clients, so modules should not be called until the module is ready to read content.

```
$ret = $r->should_client_block();
```

- **arg1: \$r (Apache::RequestRec)**

The current request

- **ret: \$ret (integer)**

0 if there is no message to read, 1 otherwise

1.3.4 print

Send data to the client.

```
$ret = $r->print(@msg);
```

- **arg1: \$r (Apache::RequestRec)**
- **arg2: @msg (ARRAY)**
- **ret: \$ret (number)**

1.3.5 read

Read data from the client.

```
$read_count = $r->read($buffer, $len, $offset);
```

META: same as CORE::read, minus the filehandle argument

- **arg1: \$r (Apache::RequestRec)**
- **arg2: \$buffer (scalar)**
- **arg3: \$len (scalar)**
- **arg4: \$offset (number)**
- **ret: \$read_count (number)**

How many characters were actually read

1.3.6 rflush

Flush any buffered data to the client.

```
$ret = $r->rflush();
```

- **arg1: \$r (Apache::RequestRec)**
- **ret: \$ret (integer)**

Unless `$| > 0`, data sent via `$r->print()` is buffered. This method flushes that data to the client.

1.3.7 sendfile

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->sendfile($filename, $offset, $len);
```

- **arg1: \$r (Apache::RequestRec)**
- **arg2: \$filename (Apache::RequestRec)**
- **arg3: \$offset (string)**
- **arg4: \$len (integer)**
- **ret: \$ret (integer)**

1.3.8 write

META: Autogenerated - needs to be reviewed/completed

Write data to the client

```
$ret = $r->write($buffer, $bufsiz, $offset);
```

- **arg1: \$r** (**Apache::RequestRec**)
- **arg2: \$buffer** (scalar)
- **arg3: \$bufsiz** (scalar)
- **arg4: \$offset** (number)
- **ret: \$ret** (number)

1.4 TIE Interface

1.4.1 OPEN

META: Autogenerated - needs to be reviewed/completed

```
$ret = OPEN($self, $arg1, $arg2);
```

- **arg1: \$self** (scalar)
- **arg2: \$arg1** (scalar)
- **arg3: \$arg2** (scalar)
- **ret: \$ret** (integer)

1.4.2 UNTIE

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->UNTIE($refcnt);
```

- **arg1: \$r** (**Apache::RequestRec**)
- **arg2: \$refcnt** (**Apache::RequestRec**)
- **ret: \$ret** (scalar)

1.4.3 PRINTF

META: Autogenerated - needs to be reviewed/completed

```
$ret = PRINTF(...);
```

- **arg1: ...** (scalar)
- **ret: \$ret** (number)

1.4.4 CLOSE

META: Autogenerated - needs to be reviewed/completed

1.4.5 PRINT

```
$ret = $r->CLOSE();
```

- **arg1:** `$r` (`Apache::RequestRec`)
- **ret:** `$ret` (scalar)

1.4.5 PRINT

META: Autogenerated - needs to be reviewed/completed

```
$ret = PRINT(...);
```

- **arg1:** `...` (scalar)
- **ret:** `$ret` (number)

1.4.6 BINMODE

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->BINMODE();
```

- **arg1:** `$r` (`Apache::RequestRec`)
- **ret:** `$ret` (scalar)

1.4.7 WRITE

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->WRITE($buffer, $bufsiz, $offset);
```

- **arg1:** `$r` (`Apache::RequestRec`)
- **arg2:** `$buffer` (scalar)
- **arg3:** `$bufsiz` (scalar)
- **arg4:** `$offset` (integer)
- **ret:** `$ret` (integer)

1.4.8 TIEHANDLE

META: Autogenerated - needs to be reviewed/completed

```
$ret = TIEHANDLE($stashsv, $sv);
```

- **arg1:** `$stashsv` (scalar)
- **arg2:** `$sv` (scalar)
- **ret:** `$ret` (scalar)

1.4.9 READ

META: Autogenerated - needs to be reviewed/completed

```
$ret = $r->READ($buffer, $len, $offset);
```

- **arg1: \$r (Apache::RequestRec)**
- **arg2: \$buffer (scalar)**
- **arg3: \$len (scalar)**
- **arg4: \$offset (integer)**
- **ret: \$ret (scalar)**

1.5 See Also

mod_perl 2.0 documentation.

1.6 Copyright

mod_perl 2.0 and its core modules are copyrighted under The Apache Software License, Version 1.1.

1.7 Authors

The mod_perl development team and numerous contributors.

Table of Contents:

1	Apache::RequestIO - Perl API for Apache request record IO	1
1.1	Synopsis	2
1.2	Description	2
1.3	API	2
1.3.1	discard_request_body	2
1.3.2	setup_client_block	2
1.3.3	should_client_block	3
1.3.4	print	3
1.3.5	read	4
1.3.6	rflush	4
1.3.7	sendfile	4
1.3.8	write	4
1.4	TIE Interface	5
1.4.1	OPEN	5
1.4.2	UNTIE	5
1.4.3	PRINTF	5
1.4.4	CLOSE	5
1.4.5	PRINT	6
1.4.6	BINMODE	6
1.4.7	WRITE	6
1.4.8	TIEHANDLE	6
1.4.9	READ	7
1.5	See Also	7
1.6	Copyright	7
1.7	Authors	7