

# **1 APR::Socket - Perl API for XXX**

## 1.1 Synopsis

```
use APR::Socket ();
```

META: to be completed

## 1.2 Description

META: to be completed

## 1.3 API

`APR::Socket` provides the following functions and/or methods:

### 1.3.1 *bind*

META: Autogenerated - needs to be reviewed/completed

Bind the socket to its associated port

```
$ret = $sock->bind($sa);
```

- **arg1: \$sock (APR::Socket)**

The socket to bind

- **arg2: \$sa (APR::SockAddr)**

The socket address to bind to

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

This may be where we will find out if there is any other process using the selected port.

### 1.3.2 *close*

META: Autogenerated - needs to be reviewed/completed

Close a socket.

```
$ret = $thesocket->close();
```

- **arg1: \$thesocket (APR::Socket)**

The socket to close

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

### ***1.3.3 connect***

META: Autogenerated - needs to be reviewed/completed

Issue a connection request to a socket either on the same machine or a different one.

```
$ret = $sock->connect($sa);
```

- **arg1: \$sock (APR::Socket)**

The socket we wish to use for our side of the connection

- **arg2: \$sa (APR::SockAddr)**

The address of the machine we wish to connect to. If NULL, APR assumes that the sockaddr\_in in the apr\_socket is completely filled out.

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

### ***1.3.4 listen***

META: Autogenerated - needs to be reviewed/completed

Listen to a bound socket for connections.

```
$ret = $sock->listen($backlog);
```

- **arg1: \$sock (APR::Socket)**

The socket to listen on

- **arg2: \$backlog (integer)**

The number of outstanding connections allowed in the sockets listen queue. If this value is less than zero, the listen queue size is set to zero.

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

### ***1.3.5 opt\_get***

META: Autogenerated - needs to be reviewed/completed

Query socket options for the specified socket

### 1.3.6 opt\_set

```
$ret = $sock->opt_get($opt, $on);
```

- **arg1: \$sock (APR::Socket)**

The socket to query

- **arg2: \$opt (integer)**

The option we would like to query. One of:

```
APR::SO_DEBUG          -- turn on debugging information
APR::SO_KEEPALIVE      -- keep connections active
APR::SO_LINGER         -- lingers on close if data is present
APR::SO_NONBLOCK       -- Turns blocking on/off for socket
APR::SO_REUSEADDR      -- The rules used in validating addresses
                        supplied to bind should allow reuse
                        of local addresses.
APR::SO_SNDBUF         -- Set the SendBufferSize
APR::SO_RCVBUF         -- Set the ReceiveBufferSize
APR::SO_DISCONNECTED -- Query the disconnected state of the socket.
                        (Currently only used on Windows)
```

- **arg3: \$on (integer)**

Socket option returned on the call.

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

## 1.3.6 opt\_set

META: Autogenerated - needs to be reviewed/completed

Setup socket options for the specified socket

```
$ret = $sock->opt_set($opt, $on);
```

- **arg1: \$sock (APR::Socket)**

The socket to set up.

- **arg2: \$opt (integer)**

The option we would like to configure. One of:

```
APR::SO_DEBUG          -- turn on debugging information
APR::SO_KEEPALIVE      -- keep connections active
APR::SO_LINGER         -- lingers on close if data is present
APR::SO_NONBLOCK       -- Turns blocking on/off for socket
APR::SO_REUSEADDR      -- The rules used in validating addresses
                        supplied to bind should allow reuse of local
                        addresses.
APR::SO_SNDBUF         -- Set the SendBufferSize
APR::SO_RCVBUF         -- Set the ReceiveBufferSize
```

- **arg3: \$on (integer)**

Value for the option.

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

### ***1.3.7 recvfrom***

META: Autogenerated - needs to be reviewed/completed

```
$ret = $from->recvfrom($sock, $flags, $buf, $len);
```

- **arg1: \$from (APR::SockAddr)**

The apr\_sockaddr\_t to fill in the recipient info

- **arg2: \$sock (APR::SockAddr)**

The socket to use

- **arg3: \$flags (integer)**

The flags to use

- **arg4: \$buf (integer)**

The buffer to use

- **arg5: \$len (string)**

The length of the available buffer

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

### ***1.3.8 sendto***

META: Autogenerated - needs to be reviewed/completed

```
$ret = $sock->sendto($where, $flags, $buf, $len);
```

- **arg1: \$sock (APR::Socket)**

The socket to send from

- **arg2: \$where (APR::Socket)**

The apr\_sockaddr\_t describing where to send the data

## 1.4 See Also

- **arg3: \$flags (integer)**

The flags to use

- **arg4: \$buf (scalar)**

The data to send

- **arg5: \$len (string)**

The length of the data to send

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

### 1.3.9 *timeout\_set*

META: Autogenerated - needs to be reviewed/completed

Setup socket timeout for the specified socket

```
$ret = $sock->timeout_set($t);
```

- **arg1: \$sock (APR::Socket)**

The socket to set up.

- **arg2: \$t (number)**

Value for the timeout:

```
t > 0  -- read and write calls return APR::TIMEUP if specified time
        elapses with no data read or written
t == 0 -- read and write calls never block
t < 0  -- read and write calls block
```

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

## 1.4 See Also

mod\_perl 2.0 documentation.

## 1.5 Copyright

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

## 1.6 Authors

The mod\_perl development team and numerous contributors.





## Table of Contents:

1	APR::Socket - Perl API for XXX	1
1.1	Synopsis	2
1.2	Description	2
1.3	API	2
1.3.1	bind	2
1.3.2	close	2
1.3.3	connect	3
1.3.4	listen	3
1.3.5	opt_get	3
1.3.6	opt_set	4
1.3.7	recvfrom	5
1.3.8	sendto	5
1.3.9	timeout_set	6
1.4	See Also	6
1.5	Copyright	6
1.6	Authors	7