



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'APR::IpSubnet.3pm'***

***\$ man APR::IpSubnet.3pm***

libapache2-mod-perl2-2.0.12::doUserConlibapache2-mod-perl2-2.0.12::docs::api::APR::IpSubnet(3pm)

NAME

APR::IpSubnet - Perl API for accessing APRs ip\_subnet structures

Synopsis

```
use APR::IpSubnet ();

my $ipsub = APR::IpSubnet->new($pool, "127.0.0.1");

 Sok = $ipsub->test($sock_addr);
```

Description

"APR::IpSubnet" object represents a range of IP addresses (IPv4/IPv6). A socket connection can be matched against this range to test whether the IP it's coming from is inside or outside of this range.

API

"APR::IpSubnet" provides the following functions and/or methods:

"new"

Create an IP subnet representation object

```
$ipsubnet = APR::IpSubnet->new($pool, $ip);
```

```
$ipsubnet = APR::IpSubnet->new($pool, $ip, $mask_or_numbits);
```

obj: "APR::IpSubnet" (class name)

arg1: \$pool ( "APR::Pool object" )

arg2: \$ip ( string )

IP address in one of the two formats: IPv4 (e.g. "127.0.0.1") or IPv6 (e.g. "::1").

IPv6 addresses are accepted only if APR has the IPv6 support enabled.

opt arg3: \$mask\_or\_numbits ( string )

An optional IP mask (e.g. "255.0.0.0") or number of bits (e.g. "15").

If none provided, the default is not to mask off.

ret: \$ret ( "APR::IpSubnet object" )

The IP-subnet object

excpt: "APR::Error"

since: 2.0.00

"test"

Test the IP address in the socket address object against a pre-built ip-subnet representation.

```
$ret = $ipsub->test($sockaddr);
```

obj: \$ipsub ( "APR::IpSubnet object" )

The ip-subnet representation

arg1: \$sockaddr ( "APR::SockAddr object" )

The socket address to test

ret: \$ret ( boolean )

true if the socket address is within the subnet, false otherwise

since: 2.0.00

This method is used for testing whether or not an address is within a subnet. It's used by module "mod\_access" to check whether the client IP fits into the IP range, supplied by "Allow"/"Deny" directives.

Example:

Allow accesses only from the localhost (IPv4):

```
use APR::IpSubnet ();
use Apache2::Connection ();
use Apache2::RequestRec ();
my $ipsub = APR::IpSubnet->new($r->pool, "127.0.0.1");
ok $ipsub->test($r->connection->remote_addr);
```

See Also

mod\_perl 2.0 documentation.

Copyright

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

Authors

The mod\_perl development team and numerous contributors.

perl v5.34.0

libapache2-mod-perl2-2.0.12::docs::api::APR::IpSubnet(3pm)