

**NAME**

IO::Socket::UNIX - Object interface for AF\_UNIX domain sockets

**SYNOPSIS**

```
use IO::Socket::UNIX;

my $SOCK_PATH = "$ENV{HOME}/unix-domain-socket-test.sock";

# Server:
my $server = IO::Socket::UNIX->new(
    Type => SOCK_STREAM(),
    Local => $SOCK_PATH,
    Listen => 1,
);

my $count = 1;
while (my $conn = $server->accept()) {
    $conn->print("Hello " . ($count++) . "\n");
}

# Client:
my $client = IO::Socket::UNIX->new(
    Type => SOCK_STREAM(),
    Peer => $SOCK_PATH,
);

# Now read and write from $client
```

**DESCRIPTION**

[IO::Socket::UNIX](#) provides an object interface to creating and using sockets in the AF\_UNIX domain. It is built upon the [IO::Socket](#) interface and inherits all the methods defined by [IO::Socket](#).

**CONSTRUCTOR**

`new ( [ARGS] )`

Creates an [IO::Socket::UNIX](#) object, which is a reference to a newly created symbol (see the [Symbol](#) package). `new` optionally takes arguments, these arguments are in key-value pairs.

In addition to the key-value pairs accepted by [IO::Socket](#), [IO::Socket::UNIX](#) provides.

```
Type Type of socket (eg SOCK_STREAM or SOCK_DGRAM)
Local Path to local fifo
Peer Path to peer fifo
Listen Queue size for listen
```

If the constructor is only passed a single argument, it is assumed to be a `Peer` specification.

If the `Listen` argument is given, but false, the queue size will be set to 5.

**METHODS**

*hostpath()*

Returns the pathname to the fifo at the local end

*peerpath()*

Returns the pathanme to the fifo at the peer end

**SEE ALSO**

Socket, [IO::Socket](#)

**AUTHOR**

Graham Barr. Currently maintained by the Perl Porters. Please report all bugs to <perlbug@perl.org>.

**COPYRIGHT**

Copyright (c) 1996-8 Graham Barr <gbarr@pobox.com>. All rights reserved. This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.