

NAME

BIO_socket, BIO_bind, BIO_connect, BIO_listen, BIO_accept_ex, BIO_closesocket - BIO socket communication setup routines

SYNOPSIS

```
#include <openssl/bio.h>
```

```
int BIO_socket(int domain, int socktype, int protocol, int options);
int BIO_bind(int sock, const BIO_ADDR *addr, int options);
int BIO_connect(int sock, const BIO_ADDR *addr, int options);
int BIO_listen(int sock, const BIO_ADDR *addr, int options);
int BIO_accept_ex(int accept_sock, BIO_ADDR *peer, int options);
int BIO_closesocket(int sock);
```

DESCRIPTION

BIO_socket() creates a socket in the domain **domain**, of type **socktype** and **protocol**. Socket **options** are currently unused, but is present for future use.

BIO_bind() binds the source address and service to a socket and may be useful before calling **BIO_connect()**. The options may include **BIO_SOCKET_REUSEADDR**, which is described in "FLAGS" below.

BIO_connect() connects **sock** to the address and service given by **addr**. Connection **options** may be zero or any combination of **BIO_SOCKET_KEEPALIVE**, **BIO_SOCKET_NONBLOCK** and **BIO_SOCKET_NODELAY**. The flags are described in "FLAGS" below.

BIO_listen() has **sock** start listening on the address and service given by **addr**. Connection **options** may be zero or any combination of **BIO_SOCKET_KEEPALIVE**, **BIO_SOCKET_NONBLOCK**, **BIO_SOCKET_NODELAY**, **BIO_SOCKET_REUSEADDR** and **BIO_SOCKET_V6_ONLY**. The flags are described in "FLAGS" below.

BIO_accept_ex() waits for an incoming connections on the given socket **accept_sock**. When it gets a connection, the address and port of the peer gets stored in **peer** if that one is non-NULL. Accept **options** may be zero or **BIO_SOCKET_NONBLOCK**, and is applied on the accepted socket. The flags are described in "FLAGS" below.

BIO_closesocket() closes **sock**.

FLAGS

BIO_SOCKET_KEEPALIVE

Enables regular sending of keep-alive messages.

BIO_SOCK_NONBLOCK

Sets the socket to nonblocking mode.

BIO_SOCK_NODELAY

Corresponds to **TCP_NODELAY**, and disables the Nagle algorithm. With this set, any data will be sent as soon as possible instead of being buffered until there's enough for the socket to send out in one go.

BIO_SOCK_REUSEADDR

Try to reuse the address and port combination for a recently closed port.

BIO_SOCK_V6_ONLY

When creating an IPv6 socket, make it only listen for IPv6 addresses and not IPv4 addresses mapped to IPv6.

These flags are bit flags, so they are to be combined with the "|" operator, for example:

```
BIO_connect(sock, addr, BIO_SOCK_KEEPAIVE | BIO_SOCK_NONBLOCK);
```

RETURN VALUES

BIO_socket() returns the socket number on success or **INVALID_SOCKET** (-1) on error. When an error has occurred, the OpenSSL error stack will hold the error data and `errno` has the system error.

BIO_bind(), **BIO_connect()** and **BIO_listen()** return 1 on success or 0 on error. When an error has occurred, the OpenSSL error stack will hold the error data and `errno` has the system error.

BIO_accept_ex() returns the accepted socket on success or **INVALID_SOCKET** (-1) on error. When an error has occurred, the OpenSSL error stack will hold the error data and `errno` has the system error.

SEE ALSO

BIO_ADDR(3)

HISTORY

BIO_gethostname(), **BIO_get_port()**, **BIO_get_host_ip()**, **BIO_get_accept_socket()** and **BIO_accept()** were deprecated in OpenSSL 1.1.0. Use the functions described above instead.

COPYRIGHT

Copyright 2016-2020 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.