NAME

BIO_s_connect, BIO_new_connect, BIO_set_conn_hostname, BIO_set_conn_port, BIO_set_conn_address, BIO_set_conn_ip_family, BIO_get_conn_hostname, BIO_get_conn_port, BIO_get_conn_address, BIO_get_conn_ip_family, BIO_set_nbio, BIO_do_connect - connect BIO

SYNOPSIS

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

const BIO_METHOD *BIO_s_connect(void);

BIO *BIO_new_connect(const char *name);

long BIO_set_conn_hostname(BIO *b, char *name);
long BIO_set_conn_port(BIO *b, char *port);
long BIO_set_conn_address(BIO *b, BIO_ADDR *addr);
long BIO_set_conn_ip_family(BIO *b, long family);
const char *BIO_get_conn_hostname(BIO *b);
const char *BIO_get_conn_port(BIO *b);
const BIO_ADDR *BIO_get_conn_address(BIO *b);
const long BIO_get_conn_ip_family(BIO *b);

long BIO_set_nbio(BIO *b, long n);

long BIO_do_connect(BIO *b);
```

DESCRIPTION

BIO_s_connect() returns the connect BIO method. This is a wrapper round the platform's TCP/IP socket connection routines.

Using connect BIOs, TCP/IP connections can be made and data transferred using only BIO routines. In this way any platform specific operations are hidden by the BIO abstraction.

Read and write operations on a connect BIO will perform I/O on the underlying connection. If no connection is established and the port and hostname (see below) is set up properly then a connection is established first.

Connect BIOs support **BIO_puts()** but not **BIO_gets()**.

If the close flag is set on a connect BIO then any active connection is shutdown and the socket closed when the BIO is freed.

Calling **BIO_reset()** on a connect BIO will close any active connection and reset the BIO into a state where it can connect to the same host again.

BIO_new_connect() combines **BIO_new()** and **BIO_set_conn_hostname()** into a single call: that is it creates a new connect BIO with hostname **name**.

BIO_set_conn_hostname() uses the string **name** to set the hostname. The hostname can be an IP address; if the address is an IPv6 one, it must be enclosed with brackets "[" and "]". The hostname can also include the port in the form hostname:port; see **BIO_parse_hostserv**(3) and **BIO_set_conn_port()** for details.

BIO_set_conn_port() sets the port to **port**. **port** can be the numerical form or a service string such as "http", which will be mapped to a port number using the system function **getservbyname()**.

BIO_set_conn_address() sets the address and port information using a **BIO_ADDR**(3ssl).

BIO_set_conn_ip_family() sets the IP family.

BIO_get_conn_hostname() returns the hostname of the connect BIO or NULL if the BIO is initialized but no hostname is set. This return value is an internal pointer which should not be modified.

BIO_get_conn_port() returns the port as a string. This return value is an internal pointer which should not be modified.

BIO_get_conn_address() returns the address information as a BIO_ADDR. This return value is an internal pointer which should not be modified.

BIO_get_conn_ip_family() returns the IP family of the connect BIO.

BIO_set_nbio() sets the non blocking I/O flag to **n**. If **n** is zero then blocking I/O is set. If **n** is 1 then non blocking I/O is set. Blocking I/O is the default. The call to **BIO_set_nbio()** should be made before the connection is established because non blocking I/O is set during the connect process.

BIO_do_connect() attempts to connect the supplied BIO. This performs an SSL/TLS handshake as far as supported by the BIO. For non-SSL BIOs the connection is done typically at TCP level. If domain name resolution yields multiple IP addresses all of them are tried after **connect()** failures. The function returns 1 if the connection was established successfully. A zero or negative value is returned if the connection could not be established. The call **BIO_should_retry()** should be used for non blocking connect BIOs to determine if the call should be retried. If a connection has already been established this call has no effect.

NOTES

If blocking I/O is set then a non positive return value from any I/O call is caused by an error condition, although a zero return will normally mean that the connection was closed.

If the port name is supplied as part of the hostname then this will override any value set with **BIO_set_conn_port()**. This may be undesirable if the application does not wish to allow connection to arbitrary ports. This can be avoided by checking for the presence of the ':' character in the passed hostname and either indicating an error or truncating the string at that point.

The values returned by **BIO_get_conn_hostname()**, **BIO_get_conn_address()**, and **BIO_get_conn_port()** are updated when a connection attempt is made. Before any connection attempt the values returned are those set by the application itself.

Applications do not have to call **BIO_do_connect()** but may wish to do so to separate the connection process from other I/O processing.

If non blocking I/O is set then retries will be requested as appropriate.

It addition to **BIO_should_read()** and **BIO_should_write()** it is also possible for **BIO_should_io_special()** to be true during the initial connection process with the reason BIO_RR_CONNECT. If this is returned then this is an indication that a connection attempt would block, the application should then take appropriate action to wait until the underlying socket has connected and retry the call.

BIO_set_conn_hostname(), **BIO_set_conn_port()**, **BIO_get_conn_hostname()**, **BIO_set_conn_address()**, **BIO_get_conn_port()**, **BIO_get_conn_address()**, **BIO_set_conn_ip_family()**, **BIO_get_conn_ip_family()**, **BIO_set_nbio()**, and **BIO_do_connect()** are macros.

RETURN VALUES

BIO_s_connect() returns the connect BIO method.

BIO_set_conn_address(), **BIO_set_conn_port()**, and **BIO_set_conn_ip_family()** return 1 or <=0 if an error occurs.

BIO_set_conn_hostname() returns 1 on success and <=0 on failure.

BIO_get_conn_address() returns the address information or NULL if none was set.

BIO_get_conn_hostname() returns the connected hostname or NULL if none was set.

BIO_get_conn_ip_family() returns the address family or -1 if none was set.

BIO_get_conn_port() returns a string representing the connected port or NULL if not set.

BIO_set_nbio() returns 1 or <=0 if an error occurs.

BIO_do_connect() returns 1 if the connection was successfully established and <=0 if the connection failed.

EXAMPLES

This is example connects to a webserver on the local host and attempts to retrieve a page and copy the result to standard output.

```
BIO *cbio, *out;
int len;
char tmpbuf[1024];

cbio = BIO_new_connect("localhost:http");
out = BIO_new_fp(stdout, BIO_NOCLOSE);
if (BIO_do_connect(cbio) <= 0) {
    fprintf(stderr, "Error connecting to server\n");
    ERR_print_errors_fp(stderr);
    exit(1);
}
BIO_puts(cbio, "GET / HTTP/1.0\n\n");
for (;;) {
    len = BIO_read(cbio, tmpbuf, 1024);
    if (len <= 0)
        break;
    BIO_write(out, tmpbuf, len);
}
BIO_free(cbio);
BIO_free(out);
```

SEE ALSO

BIO_ADDR(3), **BIO_parse_hostserv**(3)

HISTORY

BIO_set_conn_int_port(), **BIO_get_conn_int_port()**, **BIO_set_conn_ip()**, and **BIO_get_conn_ip()** were removed in OpenSSL 1.1.0. Use **BIO_set_conn_address()** and **BIO_get_conn_address()** instead.

COPYRIGHT