

NAME

`SSL_connect` - initiate the TLS/SSL handshake with an TLS/SSL server

SYNOPSIS

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

```
int SSL_connect(SSL *ssl);
```

DESCRIPTION

`SSL_connect()` initiates the TLS/SSL handshake with a server. The communication channel must already have been set and assigned to the `ssl` by setting an underlying **BIO**.

NOTES

The behaviour of `SSL_connect()` depends on the underlying BIO.

If the underlying BIO is **blocking**, `SSL_connect()` will only return once the handshake has been finished or an error occurred.

If the underlying BIO is **nonblocking**, `SSL_connect()` will also return when the underlying BIO could not satisfy the needs of `SSL_connect()` to continue the handshake, indicating the problem by the return value -1. In this case a call to `SSL_get_error()` with the return value of `SSL_connect()` will yield **SSL_ERROR_WANT_READ** or **SSL_ERROR_WANT_WRITE**. The calling process then must repeat the call after taking appropriate action to satisfy the needs of `SSL_connect()`. The action depends on the underlying BIO. When using a nonblocking socket, nothing is to be done, but `select()` can be used to check for the required condition. When using a buffering BIO, like a BIO pair, data must be written into or retrieved out of the BIO before being able to continue.

Many systems implement Nagle's algorithm by default which means that it will buffer outgoing TCP data if a TCP packet has already been sent for which no corresponding ACK has been received yet from the peer. This can have performance impacts after a successful TLSv1.3 handshake or a successful TLSv1.2 (or below) resumption handshake, because the last peer to communicate in the handshake is the client. If the client is also the first to send application data (as is typical for many protocols) then this data could be buffered until an ACK has been received for the final handshake message.

The **TCP_NODELAY** socket option is often available to disable Nagle's algorithm. If an application opts to disable Nagle's algorithm consideration should be given to turning it back on again later if appropriate. The helper function `BIO_set_tcp_ndelay()` can be used to turn on or off the **TCP_NODELAY** option.

RETURN VALUES

The following return values can occur:

- 0 The TLS/SSL handshake was not successful but was shut down controlled and by the specifications of the TLS/SSL protocol. Call **SSL_get_error()** with the return value **ret** to find out the reason.
- 1 The TLS/SSL handshake was successfully completed, a TLS/SSL connection has been established.
- <0 The TLS/SSL handshake was not successful, because a fatal error occurred either at the protocol level or a connection failure occurred. The shutdown was not clean. It can also occur if action is needed to continue the operation for nonblocking BIOs. Call **SSL_get_error()** with the return value **ret** to find out the reason.

SEE ALSO

SSL_get_error(3), **SSL_accept(3)**, **SSL_shutdown(3)**, **ssl(7)**, **bio(7)**, **SSL_set_connect_state(3)**, **SSL_do_handshake(3)**, **SSL_CTX_new(3)**

COPYRIGHT

Copyright 2000-2020 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>>.