NAME

SSL_do_handshake - perform a TLS/SSL handshake

SYNOPSIS

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

 int SSL_do_handshake(SSL *ssl);
```

DESCRIPTION

SSL_do_handshake() will wait for a SSL/TLS handshake to take place. If the connection is in client mode, the handshake will be started. The handshake routines may have to be explicitly set in advance using either **SSL_set_connect_state**(3) or **SSL_set_accept_state**(3).

NOTES

The behaviour of **SSL_do_handshake()** depends on the underlying BIO.

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

If the underlying BIO is **nonblocking**, **SSL_do_handshake()** will also return when the underlying BIO could not satisfy the needs of **SSL_do_handshake()** to continue the handshake. In this case a call to **SSL_get_error()** with the return value of **SSL_do_handshake()** 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_do_handshake()**. 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.

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_connect**(3), **SSL_accept**(3), **ssl**(7), **bio**(7), **SSL_set_connect_state**(3)

COPYRIGHT