NAME

    SSL_set_connect_state, SSL_set_accept_state, SSL_is_server - functions for manipulating and examining the client or server mode of an SSL object

SYNOPSIS

    #include <openssl/ssl.h>

    void SSL_set_connect_state(SSL *ssl);

    void SSL_set_accept_state(SSL *ssl);

    int SSL_is_server(const SSL *ssl);

DESCRIPTION

    SSL_set_connect_state() sets **ssl** to work in client mode.

    SSL_set_accept_state() sets **ssl** to work in server mode.

    SSL_is_server() checks if **ssl** is working in server mode.

NOTES

    When the SSL_CTX object was created with **SSL_CTX_new**(3), it was either assigned a dedicated client method, a dedicated server method, or a generic method, that can be used for both client and server connections. (The method might have been changed with **SSL_CTX_set_ssl_version**(3) or **SSL_set_ssl_method**(3).)

    When beginning a new handshake, the SSL engine must know whether it must call the connect (client) or accept (server) routines. Even though it may be clear from the method chosen, whether client or server mode was requested, the handshake routines must be explicitly set.

    When using the **SSL_connect**(3) or **SSL_accept**(3) routines, the correct handshake routines are automatically set. When performing a transparent negotiation using **SSL_write_ex**(3), **SSL_write**(3), **SSL_read_ex**(3), or **SSL_read**(3), the handshake routines must be explicitly set in advance using either **SSL_set_connect_state()** or **SSL_set_accept_state()**.

    If **SSL_is_server()** is called before **SSL_set_connect_state()** or **SSL_set_accept_state()** is called (either automatically or explicitly), the result depends on what method was used when SSL_CTX was created with **SSL_CTX_new**(3). If a generic method or a dedicated server method was passed to **SSL_CTX_new**(3), **SSL_is_server()** returns 1; otherwise, it returns 0.

RETURN VALUES

    SSL_set_connect_state() and **SSL_set_accept_state()** do not return diagnostic information.

    SSL_is_server() returns 1 if **ssl** is working in server mode or 0 for client mode.

SEE ALSO

    ssl(7), **SSL_new**(3), **SSL_CTX_new**(3), **SSL_connect**(3), **SSL_accept**(3), **SSL_write_ex**(3),
    SSL_write(3), **SSL_read_ex**(3), **SSL_read**(3), **SSL_do_handshake**(3), **SSL_CTX_set_ssl_version**(3)

COPYRIGHT