NAME

SSL_set_fd, SSL_set_rfd, SSL_set_wfd - connect the SSL object with a file descriptor

SYNOPSIS

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

int SSL_set_fd(SSL *ssl, int fd);
int SSL_set_rfd(SSL *ssl, int fd);
int SSL_set_wfd(SSL *ssl, int fd);
```

DESCRIPTION

SSL_set_fd() sets the file descriptor **fd** as the input/output facility for the TLS/SSL (encrypted) side of **ssl**. **fd** will typically be the socket file descriptor of a network connection.

When performing the operation, a **socket BIO** is automatically created to interface between the **ssl** and **fd**. The BIO and hence the SSL engine inherit the behaviour of **fd**. If **fd** is nonblocking, the **ssl** will also have nonblocking behaviour.

If there was already a BIO connected to **ssl**, **BIO_free()** will be called (for both the reading and writing side, if different).

SSL_set_rfd() and **SSL_set_wfd()** perform the respective action, but only for the read channel or the write channel, which can be set independently.

RETURN VALUES

The following return values can occur:

0 The operation failed. Check the error stack to find out why.

1 The operation succeeded.

NOTES

On Windows, a socket handle is a 64-bit data type (UINT_PTR), which leads to a compiler warning (conversion from ’SOCKET’ to ’int’, possible loss of data) when passing the socket handle to SSL_set_***fd()**. For the time being, this warning can safely be ignored, because although the Microsoft documentation claims that the upper limit is INVALID_SOCKET-1 (2^64 - 2), in practice the current **socket()** implementation returns an index into the kernel handle table, the size of which is limited to 2^24.

SEE ALSO

SSL_get_fd(3), **SSL_set_bio**(3), **SSL_connect**(3), **SSL_accept**(3), **SSL_shutdown**(3), **ssl**(7) , **bio**(7)

COPYRIGHT