NAME

SSL_CTX_set_mode, SSL_set_mode, SSL_CTX_get_mode, SSL_get_mode - manipulate SSL engine mode

SYNOPSIS

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

long SSL_CTX_set_mode(SSL_CTX *ctx, long mode);
long SSL_set_mode(SSL *ssl, long mode);

long SSL_CTX_get_mode(SSL_CTX *ctx);
long SSL_get_mode(SSL *ssl);
```

DESCRIPTION

SSL_CTX_set_mode() adds the mode set via bitmask in **mode** to **ctx**. Options already set before are not cleared.

SSL_set_mode() adds the mode set via bitmask in **mode** to **ssl**. Options already set before are not cleared.

SSL_CTX_get_mode() returns the mode set for **ctx**.

SSL_get_mode() returns the mode set for **ssl**.

NOTES

The following mode changes are available:

SSL_MODE_ENABLE_PARTIAL_WRITE

Allow SSL_write(..., n) to return r with 0 < r < n (i.e. report success when just a single record has been written). When not set (the default), *SSL_write()* will only report success once the complete chunk was written. Once*SSL_write()* returns with r , r bytes have been successfully written and the next call to *SSL_write()* must only send the n-r bytes left, imitating the behaviour of *write()*.

SSL_MODE_ACCEPT_MOVING_WRITE_BUFFER

Make it possible to retry *SSL_write()* with changed buffer location (the buffer contents must stay the same). This is not the default to avoid the misconception that non-blocking *SSL_write()* behaves like non-blocking *write()*.

SSL_MODE_AUTO_RETRY

Never bother the application with retries if the transport is blocking. If a renegotiation take place during normal operation, a *SSL_read(3)* or *SSL_write(3)* would return with -1 and indicate the need to retry with SSL_ERROR_WANT_READ. In a non-blocking environment applications must be prepared to handle incomplete read/write operations. In a blocking environment, applications are not always prepared to deal with read/write operations returning without success report. The flag SSL_MODE_AUTO_RETRY will cause read/write operations to only return after the handshake and successful completion.

SSL_MODE_RELEASE_BUFFERS

When we no longer need a read buffer or a write buffer for a given SSL, then release the memory we were using to hold it. Using this flag can save around 34k per idle SSL connection. This flag has no effect on SSL v2 connections, or on DTLS connections.

SSL_MODE_SEND_FALLBACK_SCSV

Send TLS_FALLBACK_SCSV in the ClientHello. To be set only by applications that reconnect with a downgraded protocol version; see draft-ietf-tls-downgrade-scsv-00 for details.

DO NOT ENABLE THIS if your application attempts a normal handshake. Only use this in explicit fallback retries, following the guidance in draft-ietf-tls-downgrade-scsv-00.

SSL_MODE_ASYNC

Enable asynchronous processing. TLS I/O operations may indicate a retry with SSL_ERROR_WANT_ASYNC with this mode set if an asynchronous capable engine is used to perform cryptographic operations. See *SSL_get_error(3)*.

RETURN VALUES

SSL_CTX_set_mode() and *SSL_set_mode()* return the new mode bitmask after adding **mode**.

SSL_CTX_get_mode() and *SSL_get_mode()* return the current bitmask.

SEE ALSO

ssl(3) , *SSL_read(3)* , *SSL_write(3)* , *SSL_get_error(3)*

HISTORY

SSL_MODE_ASYNC was first added to OpenSSL 1.1.0.

COPYRIGHT