NAME

SSL_CTX_sess_set_new_cb, SSL_CTX_sess_set_remove_cb, SSL_CTX_sess_set_get_cb, SSL_CTX_sess_get_new_cb, SSL_CTX_sess_get_remove_cb, SSL_CTX_sess_get_get_cb - provide callback functions for server side external session caching

SYNOPSIS

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

void SSL_CTX_sess_set_new_cb(SSL_CTX *ctx,
                             int (*new_session_cb)(SSL *, SSL_SESSION *));
void SSL_CTX_sess_set_remove_cb(SSL_CTX *ctx,
                                void (*remove_session_cb)(SSL_CTX *ctx,
                                                          SSL_SESSION *));
void SSL_CTX_sess_set_get_cb(SSL_CTX *ctx,
                             SSL_SESSION (*get_session_cb)(SSL *,
                                                           const unsigned char *,
                                                           int, int *));

int (*SSL_CTX_sess_get_new_cb(SSL_CTX *ctx))(struct ssl_st *ssl,
                                             SSL_SESSION *sess);
void (*SSL_CTX_sess_get_remove_cb(SSL_CTX *ctx))(struct ssl_ctx_st *ctx,
                                                 SSL_SESSION *sess);
SSL_SESSION *(*SSL_CTX_sess_get_get_cb(SSL_CTX *ctx))(struct ssl_st *ssl,
                                                      const unsigned char *data,
                                                      int len, int *copy);
```

DESCRIPTION

SSL_CTX_sess_set_new_cb() sets the callback function that is called whenever a new session was negotiated.

SSL_CTX_sess_set_remove_cb() sets the callback function that is called whenever a session is removed by the SSL engine. For example, this can occur because a session is considered faulty or has become obsolete because of exceeding the timeout value.

SSL_CTX_sess_set_get_cb() sets the callback function that is called whenever a TLS client proposed to resume a session but the session could not be found in the internal session cache (see **SSL_CTX_set_session_cache_mode**(3)). (TLS server only.)

SSL_CTX_sess_get_new_cb(), **SSL_CTX_sess_get_remove_cb()**, and **SSL_CTX_sess_get_get_cb()** retrieve the function pointers set by the corresponding set callback functions. If a callback function has

not been set, the NULL pointer is returned.

NOTES

In order to allow external session caching, synchronization with the internal session cache is realized via callback functions. Inside these callback functions, session can be saved to disk or put into a database using the **d2i_SSL_SESSION**(3) interface.

The **new_session_cb()** is called whenever a new session has been negotiated and session caching is enabled (see **SSL_CTX_set_session_cache_mode**(3)). The **new_session_cb()** is passed the **ssl** connection and the nascent ssl session **sess**. Since sessions are reference-counted objects, the reference count on the session is incremented before the callback, on behalf of the application. If the callback returns **0**, the session will be immediately removed from the internal cache and the reference count released. If the callback returns **1**, the application retains the reference (for an entry in the application-maintained "external session cache"), and is responsible for calling **SSL_SESSION_free()** when the session reference is no longer in use.

Note that in TLSv1.3, sessions are established after the main handshake has completed. The server decides when to send the client the session information and this may occur some time after the end of the handshake (or not at all). This means that applications should expect the **new_session_cb()** function to be invoked during the handshake (for <= TLSv1.2) or after the handshake (for TLSv1.3). It is also possible in TLSv1.3 for multiple sessions to be established with a single connection. In these case the **new_session_cb()** function will be invoked multiple times.

In TLSv1.3 it is recommended that each SSL_SESSION object is only used for resumption once. One way of enforcing that is for applications to call **SSL_CTX_remove_session**(3) after a session has been used.

The **remove_session_cb()** is called whenever the SSL engine removes a session from the internal cache. This can happen when the session is removed because it is expired or when a connection was not shutdown cleanly. It also happens for all sessions in the internal session cache when **SSL_CTX_free**(3) is called. The **remove_session_cb()** is passed the **ctx** and the ssl session **sess**. It does not provide any feedback.

The **get_session_cb()** is only called on SSL/TLS servers, and is given the session id proposed by the client. The **get_session_cb()** is always called, even when session caching was disabled. The **get_session_cb()** is passed the **ssl** connection and the session id of length **length** at the memory location **data**. By setting the parameter **copy** to **1**, the callback can require the SSL engine to increment the reference count of the SSL_SESSION object; setting **copy** to **0** causes the reference count to remain unchanged. If the **get_session_cb()** does not write to **copy**, the reference count is incremented and the session must be explicitly freed with **SSL_SESSION_free**(3).

RETURN VALUES

SSL_CTX_sess_get_new_cb(), **SSL_CTX_sess_get_remove_cb()** and **SSL_CTX_sess_get_get_cb()** return different callback function pointers respectively.

SEE ALSO

ssl(7), **d2i_SSL_SESSION**(3), **SSL_CTX_set_session_cache_mode**(3), **SSL_CTX_flush_sessions**(3), **SSL_SESSION_free**(3), **SSL_CTX_free**(3)

COPYRIGHT