

**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);
```

```
int (*new_session_cb)(struct ssl_st *ssl, SSL_SESSION *sess);
void (*remove_session_cb)(struct ssl_ctx_st *ctx, SSL_SESSION *sess);
SSL_SESSION *(*get_session_cb)(struct ssl_st *ssl, unsigned char *data, int len, int *copy);
```

**DESCRIPTION**

*SSL\_CTX\_sess\_set\_new\_cb()* sets the callback function, which is automatically called whenever a new session was negotiated.

*SSL\_CTX\_sess\_set\_remove\_cb()* sets the callback function, which is automatically called whenever a session is removed by the SSL engine, because it is considered faulty or the session has become obsolete because of exceeding the timeout value.

*SSL\_CTX\_sess\_set\_get\_cb()* sets the callback function which is called, whenever a SSL/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\(3\)](#)). (SSL/TLS server only.)

*SSL\_CTX\_sess\_get\_new\_cb()*, *SSL\_CTX\_sess\_get\_remove\_cb()*, and *SSL\_CTX\_sess\_get\_get\_cb()* allow to retrieve the function pointers of the provided 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 ssl session **sess**. If the callback returns **0**, the session will be immediately removed again.

The *remove\_session\_cb()* is called, whenever the SSL engine removes a session from the internal cache. This happens 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 with the session id proposed by the client. The *get\_session\_cb()* is always called, also when session caching was disabled. The *get\_session\_cb()* is passed the **ssl** connection, the session id of length **length** at the memory location **data**. With the parameter **copy** the callback can require the SSL engine to increment the reference count of the SSL\_SESSION object, Normally the reference count is not incremented and therefore the session must not be explicitly freed with [SSL\\_SESSION\\_free\(3\)](#).

**SEE ALSO**

*ssl(3)*, *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**

Copyright 2001-2016 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the OpenSSL license (the “License”). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.