

**NAME**

SSL\_SESSION\_new, SSL\_SESSION\_dup, SSL\_SESSION\_up\_ref, SSL\_SESSION\_free - create, free and manage SSL\_SESSION structures

**SYNOPSIS**

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

```
SSL_SESSION *SSL_SESSION_new(void);  
SSL_SESSION *SSL_SESSION_dup(const SSL_SESSION *src);  
int SSL_SESSION_up_ref(SSL_SESSION *ses);  
void SSL_SESSION_free(SSL_SESSION *session);
```

**DESCRIPTION**

**SSL\_SESSION\_new()** creates a new SSL\_SESSION structure and returns a pointer to it.

**SSL\_SESSION\_dup()** creates a new SSL\_SESSION structure that is a copy of **src**. The copy is not owned by any cache that **src** may have been in.

**SSL\_SESSION\_up\_ref()** increments the reference count on the given SSL\_SESSION structure.

**SSL\_SESSION\_free()** decrements the reference count of **session** and removes the **SSL\_SESSION** structure pointed to by **session** and frees up the allocated memory, if the reference count has reached 0. If **session** is NULL nothing is done.

**NOTES**

SSL\_SESSION objects are allocated, when a TLS/SSL handshake operation is successfully completed. Depending on the settings, see **SSL\_CTX\_set\_session\_cache\_mode(3)**, the SSL\_SESSION objects are internally referenced by the SSL\_CTX and linked into its session cache. SSL objects may be using the SSL\_SESSION object; as a session may be reused, several SSL objects may be using one SSL\_SESSION object at the same time. It is therefore crucial to keep the reference count (usage information) correct and not delete a SSL\_SESSION object that is still used, as this may lead to program failures due to dangling pointers. These failures may also appear delayed, e.g. when an SSL\_SESSION object was completely freed as the reference count incorrectly became 0, but it is still referenced in the internal session cache and the cache list is processed during a **SSL\_CTX\_flush\_sessions(3)** operation.

**SSL\_SESSION\_free()** must only be called for SSL\_SESSION objects, for which the reference count was explicitly incremented (e.g. by calling **SSL\_get1\_session()**, see **SSL\_get\_session(3)**) or when the SSL\_SESSION object was generated outside a TLS handshake operation, e.g. by using **d2i\_SSL\_SESSION(3)**. It must not be called on other SSL\_SESSION objects, as this would cause

incorrect reference counts and therefore program failures.

## RETURN VALUES

SSL\_SESSION\_new returns a pointer to the newly allocated SSL\_SESSION structure or NULL on error.

SSL\_SESSION\_dup returns a pointer to the new copy or NULL on error.

SSL\_SESSION\_up\_ref returns 1 on success or 0 on error.

## SEE ALSO

ssl(7), SSL\_get\_session(3), SSL\_CTX\_set\_session\_cache\_mode(3), SSL\_CTX\_flush\_sessions(3), d2i\_SSL\_SESSION(3)

## HISTORY

The **SSL\_SESSION\_dup**(0) function was added in OpenSSL 1.1.1.

## COPYRIGHT

Copyright 2000-2023 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (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>>.