

**NAME**

SSL\_dup, SSL\_new, SSL\_up\_ref - create an SSL structure for a connection

**SYNOPSIS**

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

```
SSL *SSL_dup(SSL *s);  
SSL *SSL_new(SSL_CTX *ctx);  
int SSL_up_ref(SSL *s);
```

**DESCRIPTION**

**SSL\_new()** creates a new **SSL** structure which is needed to hold the data for a TLS/SSL connection. The new structure inherits the settings of the underlying context **ctx**: connection method, options, verification settings, timeout settings. An **SSL** structure is reference counted. Creating an **SSL** structure for the first time increments the reference count. Freeing it (using **SSL\_free**) decrements it. When the reference count drops to zero, any memory or resources allocated to the **SSL** structure are freed.

**SSL\_up\_ref()** increments the reference count for an existing **SSL** structure.

The function **SSL\_dup()** creates and returns a new **SSL** structure from the same **SSL\_CTX** that was used to create *s*. It additionally duplicates a subset of the settings in *s* into the new **SSL** object.

For **SSL\_dup()** to work, the connection **MUST** be in its initial state and **MUST NOT** have yet started the SSL handshake. For connections that are not in their initial state **SSL\_dup()** just increments an internal reference count and returns the *same* handle. It may be possible to use **SSL\_clear(3)** to recycle an SSL handle that is not in its initial state for reuse, but this is best avoided. Instead, save and restore the session, if desired, and construct a fresh handle for each connection.

The subset of settings in *s* that are duplicated are:

- any session data if configured (including the session\_id\_context)
- any tmp\_dh settings set via **SSL\_set\_tmp\_dh(3)**, **SSL\_set\_tmp\_dh\_callback(3)**, or **SSL\_set\_dh\_auto(3)**
- any configured certificates, private keys or certificate chains
- any configured signature algorithms, or client signature algorithms
- any DANE settings
- any Options set via **SSL\_set\_options(3)**
- any Mode set via **SSL\_set\_mode(3)**
- any minimum or maximum protocol settings set via **SSL\_set\_min\_proto\_version(3)** or **SSL\_set\_max\_proto\_version(3)** (Note: Only from OpenSSL 1.1.1h and above)
- any verify mode, callback or depth set via **SSL\_set\_verify(3)** or **SSL\_set\_verify\_depth(3)** or any

configured X509 verification parameters  
any msg callback or info callback set via **SSL\_set\_msg\_callback(3)** or **SSL\_set\_info\_callback(3)**  
any default password callback set via **SSL\_set\_default\_passwd\_cb(3)**  
any session id generation callback set via **SSL\_set\_generate\_session\_id(3)**  
any configured Cipher List  
initial accept (server) or connect (client) state  
the max cert list value set via **SSL\_set\_max\_cert\_list(3)**  
the read\_ahead value set via **SSL\_set\_read\_ahead(3)**  
application specific data set via **SSL\_set\_ex\_data(3)**  
any CA list or client CA list set via **SSL\_set0\_CA\_list(3)**, **SSL\_set0\_client\_CA\_list()** or similar functions  
any security level settings or callbacks  
any configured serverinfo data  
any configured PSK identity hint  
any configured custom extensions  
any client certificate types configured via **SSL\_set1\_client\_certificate\_types**

## RETURN VALUES

The following return values can occur:

NULL

The creation of a new SSL structure failed. Check the error stack to find out the reason.

Pointer to an SSL structure

The return value points to an allocated SSL structure.

**SSL\_up\_ref()** returns 1 for success and 0 for failure.

## SEE ALSO

**SSL\_free(3)**, **SSL\_clear(3)**, **SSL\_CTX\_set\_options(3)**, **SSL\_get\_SSL\_CTX(3)**, **ssl(7)**

## 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>.