NAME

d2i_SSL_SESSION, i2d_SSL_SESSION - convert SSL_SESSION object from/to ASN1 representation

SYNOPSIS

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

SSL_SESSION *d2i_SSL_SESSION(SSL_SESSION **a, const unsigned char **pp, long length);
int i2d_SSL_SESSION(SSL_SESSION *in, unsigned char **pp);
```

DESCRIPTION

d2i_SSL_SESSION() transforms the external ASN1 representation of an SSL/TLS session, stored as binary data at location **pp** with length **length**, into an SSL_SESSION object.

i2d_SSL_SESSION() transforms the SSL_SESSION object **in** into the ASN1 representation and stores it into the memory location pointed to by **pp**. The length of the resulting ASN1 representation is returned. If **pp** is the NULL pointer, only the length is calculated and returned.

NOTES

The SSL_SESSION object is built from several *malloc()*ed parts, it can therefore not be moved, copied or stored directly. In order to store session data on disk or into a database, it must be transformed into a binary ASN1 representation.

When using *d2i_SSL_SESSION()*, the SSL_SESSION object is automatically allocated. The reference count is 1, so that the session must be explicitly removed using *SSL_SESSION_free(3)*, unless the SSL_SESSION object is completely taken over, when being called inside the *get_session_cb()* (see *SSL_CTX_sess_set_get_cb(3)*).

SSL_SESSION objects keep internal link information about the session cache list, when being inserted into one SSL_CTX object's session cache. One SSL_SESSION object, regardless of its reference count, must therefore only be used with one SSL_CTX object (and the SSL objects created from this SSL_CTX object).

When using *i2d_SSL_SESSION()*, the memory location pointed to by **pp** must be large enough to hold the binary representation of the session. There is no known limit on the size of the created ASN1 representation, so the necessary amount of space should be obtained by first calling *i2d_SSL_SESSION()* with **pp=NULL**, and obtain the size needed, then allocate the memory and call *i2d_SSL_SESSION()* again. Note that this will advance the value contained in ***pp** so it is necessary to save a copy of the original allocation. For example: int i,j; char *p, *temp; i = i2d_SSL_SESSION(sess, NULL); p = temp = malloc(i); j = i2d_SSL_SESSION(sess, &temp); assert(i == j); assert(p+i == temp);

RETURN VALUES

d2i_SSL_SESSION() returns a pointer to the newly allocated SSL_SESSION object. In case of failure the NULL-pointer is returned and the error message can be retrieved from the error stack.

i2d_SSL_SESSION() returns the size of the ASN1 representation in bytes. When the session is not valid, **0** is returned and no operation is performed.

SEE ALSO

ssl(3), *SSL_SESSION_free(3)*, *SSL_CTX_sess_set_get_cb(3)*