libssl Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Table of Contents

Structures

SceSslMemoryPoolStats

Structure storing memory pool status

Definition

```
#include <libssl.h>
typedef struct SceSslMemoryPoolStats{
        SceSize poolSize;
        SceSize maxInuseSize;
        SceSize currentInuseSize;
        SceInt32 reserved;
} SceSslMemoryPoolStats;
```

Members

poolSize	Memory pool size specified with `sceHttpInit()`
maxInuseSize	Maximum memory size used by libhttp after `sceHttpInit()`
currentInuseSize	Size of the memory currently used by libhttp
reserved	Area for future extension

Description

It is used to store current memory pool status with `sceSslGetMemoryPoolStats()`.

See Also

`sceSslGetMemoryPoolStats()`

Library Initialization/Termination

sceSslInit

Initialize the library

Definition

```
#include <libssl.h>
SceInt32 sceSslInit (
        SceSize poolSize
)
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Not multithread safe.

Arguments

poolSize	Size of the memory pool used by the library

Return Values

Value	Hexadecimal	Description
SCE_OK	0	Normal completion
SCE_SSL_ERROR_ALREADY_INITED	0x80435020	Library has already been initialized

Description

This function initializes libssl. It must be called before calling other libssl functions or performing https communication with libhttp.

Allocate a memory pool of *poolsize* bytes from the system in this function and use it as memory pool for this library.

Examples

```
#define SSL_POOLSIZE  (150 * 1024U)
ret = sceSslInit(SSL_POOLSIZE);
if(ret < 0){
        /* Error handling */
}
```

See Also

sceSslTerm()

sceSslTerm

Terminate the library

Definition

```
#include <libssl.h>
SceInt32 sceSslTerm(
        void
)
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Not multithread safe.

Arguments

None

Return Values

Value	Hexadecimal	Description
SCE_OK	0	Normal completion
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized

Description

This function terminates libssl and releases resources that had been allocated.

See Also

`sceSslInit()`

Memory Management Functions

sceSslGetMemoryPoolStats

Retrieve memory pool status

Definition

```
#include <libssl.h>
int sceSslGetMemoryPoolStats (
        SceSslMemoryPoolStats* currentStat
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

currentStat	Memory address storing memory pool status

Return Values

Value	Hexadecimal	Description
SCE_OK	0	Normal completion
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized
SCE_SSL_ERROR_INVALID_VALUE	0x804351FE	An invalid value has been set as argument

Description

Retrieves the status of the memory pool used by libhttp. Retrieved information are the maximum size of the memory pool (the memory pool size specified with `sceSslInit()`), the maximum used size after the execution of `sceSslInit()` to the present, and the current used memory size.

See Also

`sceHttpInit()`

sceSslFreeSslCertName

Free the `SceSslCertName` object

Definition

```
#include <libssl.h>
SceInt32 sceSslFreeSslCertName (
        SceSslCertName *certName
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

certName Pointer to the `SceSslCertName` object

Return Values

Value	Hexadecimal	Description
SCE_OK	0	Normal completion
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized
SCE_SSL_ERROR_INVALID_VALUE	0x804351FE	NULL was specified for *certName*

Description

This is a function for freeing the `SceSslCertName` object. The `SceSslCertName` object can be retrieved with functions such as `sceSslGetSubjectName()` and `sceSslGetIssuerName()`.

See Also

`sceSslGetSubjectName()`, `sceSslGetIssuerName()`

Certificate Information Retrieval Functions

sceSslGetSerialNumber

Get certificate serial number

Definition

```
#include <libssl.h>
SceInt32 sceSslGetSerialNumber (
        SceSslCert *sslCert,
        const SceUChar8 **sboData,
        SceSize *sboLen
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

sslCert	Pointer to the certificate for which the serial number is to be obtained. The pointer to the certificate is obtained using the callback function set by `sceHttpsSetSslCallback()`
sboData	Specifies memory location where the starting address of the serial number will be stored. Serial numbers are stored in big-endian format
sboLen	Specifies the memory address where the length of the serial number is stored

Return Values

Value	Hexadecimal	Description
`SCE_OK`	0	Normal completion
`SCE_SSL_ERROR_BEFORE_INIT`	0x80435001	Library not initialized
`SCE_SSL_ERROR_INVALID_VALUE`	0x804351FE	NULL was specified for *sslCert*, *sboData* or *sboLen*

Description

This function is used for obtaining the serial number of a certificate object. The certificate object is used as an argument of the callback function set by `sceHttpsSetSslCallback()` in the libhttp. The memory area where the serial number is stored is freed immediately after the callback function ends, so be sure to copy the serial number to a separate memory area within the callback function if you will need to reference it later.

See Also

`sceHttpsSetSslCallback()`

sceSslGetSubjectName

Get subject name of certificate

Definition

```
#include <libssl.h>
const SceSslCertName *sceSslGetSubjectName (
        SceSslCert *sslCert
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

sslCert	Pointer to the certificate for which the subject name is to be obtained. The pointer to the certificate is obtained using the callback function set by `sceHttpsSetSslCallback()`

Return Values

If the function completes normally, a pointer to the `SceSslCertName` object is returned.

If the subject name cannot be obtained, NULL is returned.

Description

This function is used for obtaining the name object of the subject of a certificate object. The certificate object is obtained as an argument of the callback function set by `sceHttpsSetSslCallback()` in the libhttp. To obtain details on the subject name object, first get the number of subject name entries using `sceSslGetNameEntryCount()`, and then use `sceSslGetNameEntryInfo()` to get information for each name entry.

See Also

`sceHttpsSetSslCallback()`

sceSslGetIssuerName

Get information on certificate issuer

Definition

```
#include <libssl.h>
const SceSslCertName *sceSslGetIssuerName (
        SceSslCert *sslCert
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

sslCert	Pointer to the certificate for which issuer information is to be obtained. The pointer to the certificate is obtained using the callback function set by `sceHttpsSetSslCallback()`

Return Values

If the function completes normally, a pointer to the `SceSslCertName` object is returned.

If the issuer information cannot be obtained, NULL is returned.

Description

This function is used for obtaining an SSL name object for the issuer information of a certificate object. The certificate object is obtained as an argument of the callback function set by `sceHttpsSetSslCallback()` in the libhttp. To obtain details on the name object, first get the number of name entries using `sceSslGetNameEntryCount()`, and then use `sceSslGetNameEntryInfo()` to get information for each name entry.

See Also

`sceHttpsSetSslCallback()`

sceSslGetNameEntryCount

Get number of name entries

Definition

```
#include <libssl.h>
SceInt32 sceSslGetNameEntryCount (
        SceSslCertName *certName
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

certName	Pointer to the SSL name object for which the number of name entries is to be obtained

Return Values

Returns the number of name entries (positive value) included in *certName* for normal termination.

Returns one of the following error codes (negative value) for errors.

Value	Hexadecimal	Description
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized
SCE_SSL_ERROR_INVALID_VALUE	0x804351FE	NULL was specified for *certName*

Description

This function is used for obtaining the number of name entries contained in an SSL name object. The SSL name object can be obtained by using functions such as sceSslGetSubjectName() or sceSslGetIssuerName().

See Also

sceHttpsSetSslCallback()

sceSslGetNameEntryInfo

Get name entry information

Definition

```
#include <libssl.h>
SceInt32 sceSslGetNameEntryInfo (
        SceSslCertName *certName,
        SceInt32 entryNum,
        SceChar8 *oidname,
        SceSize maxOidnameLen,
        SceUChar8 *value,
        SceSize maxValueLen,
        SceSize *valueLen
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

certName	Pointer to the SSL name object for which the entry is to be obtained
entryNum	Entry number to be obtained. The number of entries stored in the SSL name object can be obtained using sceSslGetNameEntryCount()
oidname	An ASCIZ string representing the entry's object ID. If set to NULL, it signifies an unknown object ID
maxOidnameLen	Maximum length that can be stored in *oidname*
value	Memory location where the starting address of the entry values will be stored. Not in ASCIZ format
maxValueLen	Maximum length that can be stored in *value*
valueLen	Memory address where the size of *value* is stored

Return Values

Value	Hexadecimal	Description
SCE_OK	0	Normal completion
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized
SCE_SSL_ERROR_INVALID_VALUE	0x804351FE	NULL was specified for *oidname*, *value* or *valueLen*

Description

This function is used for obtaining the name entry information contained in an SSL name object. The SSL name object can be obtained by using functions such as sceSslGetSubjectName() or sceSslGetIssuerName(). The memory used to store the value fetched using this function is freed as soon as the callback function set by sceHttpsSetSslCallback() ends, so be sure to copy the value to a separate memory area within the callback function if you wish to save it for later use.

See Also

sceHttpsSetSslCallback()

sceSslGetNotAfter

Get the ending time of a certificate’s effective period

Definition

```
#include <libssl.h>
SceInt32 sceSslGetNotAfter (
        SceSslCert *sslCert,
        SceRtcTick *limit
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

sslCert	Pointer to the certificate for which the ending time of the effective period is to be obtained. The pointer to the certificate is obtained using the callback function set by `sceHttpsSetSslCallback()`
limit	Address in memory where the ending time of the effective period is to be stored

Return Values

Value	Hexadecimal	Description
`SCE_OK`	0	Normal completion
`SCE_SSL_ERROR_BEFORE_INIT`	0x80435001	Library not initialized
`SCE_SSL_ERROR_INVALID_VALUE`	0x804351FE	NULL was specified for *sslCert* or *limit*
`SCE_SSL_ERROR_INVALID_FORMAT`	0x80435108	Could not get ending time from certificate specified by *sslCert*

Description

This function is used to get the ending time for the effective period of a certificate object. The certificate object is obtained as an argument of the callback function set by `sceHttpsSetSslCallback()` in the libhttp.

See Also

`sceHttpsSetSslCallback(),sceSslGetNotBefore()`

sceSslGetNotBefore

Get starting time of a certificate's effective period

Definition

```
#include <libssl.h>
SceInt32 sceSslGetNotBefore (
        SceSslCert *sslCert,
        SceRtcTick *begin
);
```

Calling Conditions

Can be called from an interrupt handler.

Can be called from a thread (does not depend on interrupt-disabled or -enabled state)

Multithread safe.

Arguments

sslCert	Pointer to the certificate for which the starting time of the effective period is to be obtained. The pointer to the certificate is obtained using the callback function set by `sceHttpsSetSslCallback()`
begin	Address in memory where the starting time of the effective period is to be stored

Return Values

Value	Hexadecimal	Description
`SCE_OK`	0	Normal completion
`SCE_SSL_ERROR_BEFORE_INIT`	0x80435001	Library not initialized
`SCE_SSL_ERROR_INVALID_VALUE`	0x804351FE	NULL was specified for *sslCert* or *begin*
`SCE_SSL_ERROR_INVALID_FORMAT`	0x80435108	Could not get starting time from certificate specified in *sslCert*

Description

This function is used to get the starting time for the effective period of a certificate object. The certificate object is obtained as an argument of the callback function set by `sceHttpsSetSslCallback()` in the libhttp.

See Also

`sceHttpsSetSslCallback(),sceSslGetNotAfter()`

Constants

Return Codes

List of return codes returned by libssl

Definition

Value	Hexadecimal	Description
SCE_SSL_ERROR_BEFORE_INIT	0x80435001	Library not initialized
SCE_SSL_ERROR_ALREADY_INITED	0x80435020	Library has already been initialized
SCE_SSL_ERROR_OUT_OF_MEMORY	0x80435022	Could not allocate memory
SCE_SSL_ERROR_INTERNAL	0x80435026	Unknown error
SCE_SSL_ERROR_NOT_FOUND	0x80435025	Could not find specified element
SCE_SSL_ERROR_INVALID_VALUE	0x804351FE	Specified parameter was not appropriate
SCE_SSL_ERROR_INVALID_FORMAT	0x80435108	The format of the specified parameter was not appropriate