



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'SSL\_get\_servername.3ossl'***

***\$ man SSL\_get\_servername.3ossl***

SSL\_CTX\_SET\_TLSEXT\_SERVERNAME\_CALSSL\_CTX\_SET\_TLSEXT\_SERVERNAME\_CALLBACK(3ossl)

#### NAME

SSL\_CTX\_set\_tlsext\_servername\_callback,  
SSL\_CTX\_set\_tlsext\_servername\_arg, SSL\_get\_servername\_type,  
SSL\_get\_servername, SSL\_set\_tlsext\_host\_name - handle server name  
indication (SNI)

#### SYNOPSIS

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

```
long SSL_CTX_set_tlsext_servername_callback(SSL_CTX *ctx,  
int (*cb)(SSL *s, int *al, void *arg));
```

```
long SSL_CTX_set_tlsext_servername_arg(SSL_CTX *ctx, void *arg);
```

```
const char *SSL_get_servername(const SSL *s, const int type);
```

```
int SSL_get_servername_type(const SSL *s);
```

```
int SSL_set_tlsext_host_name(const SSL *s, const char *name);
```

## DESCRIPTION

The functionality provided by the `servername` callback is mostly superseded by the `ClientHello` callback, which can be set using `SSL_CTX_set_client_hello_cb()`. However, even where the `ClientHello` callback is used, the `servername` callback is still necessary in order to acknowledge the `servername` requested by the client.

`SSL_CTX_set_tlsext_servername_callback()` sets the application callback `cb` used by a server to perform any actions or configuration required based on the `servername` extension received in the incoming connection. When `cb` is `NULL`, SNI is not used.

The `servername` callback should return one of the following values:

### SSL\_TLSEXT\_ERR\_OK

This is used to indicate that the `servername` requested by the client has been accepted. Typically a server will call `SSL_set_SSL_CTX()` in the callback to set up a different configuration for the selected `servername` in this case.

### SSL\_TLSEXT\_ERR\_ALERT\_FATAL

In this case the `servername` requested by the client is not accepted and the handshake will be aborted. The value of the alert to be used should be stored in the location pointed to by the `al` parameter to the callback. By default this value is initialised to `SSL_AD_UNRECOGNIZED_NAME`.

### SSL\_TLSEXT\_ERR\_ALERT\_WARNING

If this value is returned then the `servername` is not accepted by the server. However, the handshake will continue and send a warning alert instead. The value of the alert should be stored in

the location pointed to by the `al` parameter as for

`SSL_TLSEXT_ERR_ALERT_FATAL` above. Note that TLSv1.3 does not support warning alerts, so if TLSv1.3 has been negotiated then this return value is treated the same way as `SSL_TLSEXT_ERR_NOACK`.

#### `SSL_TLSEXT_ERR_NOACK`

This return value indicates that the servername is not accepted by the server. No alerts are sent and the server will not acknowledge the requested servername.

`SSL_CTX_set_tlsext_servername_arg()` sets a context-specific argument to be passed into the callback (via the `arg` parameter) for this `SSL_CTX`.

The behaviour of `SSL_get_servername()` depends on a number of different factors. In particular note that in TLSv1.3 the servername is negotiated in every handshake. In TLSv1.2 the servername is only negotiated on initial handshakes and not on resumption handshakes.

On the client, before the handshake

If a servername has been set via a call to `SSL_set_tlsext_host_name()` then it will return that servername.

If one has not been set, but a TLSv1.2 resumption is being attempted and the session from the original handshake had a servername accepted by the server then it will return that servername.

Otherwise it returns `NULL`.

On the client, during or after the handshake and a TLSv1.2 (or below) resumption occurred

If the session from the original handshake had a servername accepted by the server then it will return that servername.

Otherwise it returns the servername set via  
SSL\_set\_tlsext\_host\_name() or NULL if it was not called.

On the client, during or after the handshake and a TLSv1.2 (or below)  
resumption did not occur

It will return the servername set via SSL\_set\_tlsext\_host\_name() or  
NULL if it was not called.

On the server, before the handshake

The function will always return NULL before the handshake

On the server, after the servername extension has been processed and a  
TLSv1.2 (or below) resumption occurred

If a servername was accepted by the server in the original  
handshake then it will return that servername, or NULL otherwise.

On the server, after the servername extension has been processed and a  
TLSv1.2 (or below) resumption did not occur

The function will return the servername requested by the client in  
this handshake or NULL if none was requested.

Note that the ClientHello callback occurs before a servername extension  
from the client is processed. The servername, certificate and ALPN  
callbacks occur after a servername extension from the client is  
processed.

SSL\_get\_servername\_type() returns the servername type or -1 if no  
servername is present. Currently the only supported type (defined in  
RFC3546) is TLSEXT\_NAMETYPE\_host\_name.

SSL\_set\_tlsext\_host\_name() sets the server name indication ClientHello  
extension to contain the value name. The type of server name indication

extension is set to `TLSEXT_NAMETYPE_host_name` (defined in RFC3546).

## NOTES

Several callbacks are executed during ClientHello processing, including the ClientHello, ALPN, and servername callbacks. The ClientHello callback is executed first, then the servername callback, followed by the ALPN callback.

The `SSL_set_tlsext_host_name()` function should only be called on SSL objects that will act as clients; otherwise the configured name will be ignored.

## RETURN VALUES

`SSL_CTX_set_tlsext_servername_callback()` and `SSL_CTX_set_tlsext_servername_arg()` both always return 1 indicating success. `SSL_set_tlsext_host_name()` returns 1 on success, 0 in case of error.

## SEE ALSO

`ssl(7)`, `SSL_CTX_set_alpn_select_cb(3)`, `SSL_get0_alpn_selected(3)`, `SSL_CTX_set_client_hello_cb(3)`

## HISTORY

`SSL_get_servername()` historically provided some unexpected results in certain corner cases. This has been fixed from OpenSSL 1.1.1e.

Prior to 1.1.1e, when the client requested a servername in an initial TLSv1.2 handshake, the server accepted it, and then the client successfully resumed but set a different explicit servername in the second handshake then when called by the client it returned the servername from the second handshake. This has now been changed to return the servername requested in the original handshake.

Also prior to 1.1.1e, if the client sent a servername in the first handshake but the server did not accept it, and then a second handshake occurred where TLSv1.2 resumption was successful then when called by the server it returned the servername requested in the original handshake. This has now been changed to NULL.

## COPYRIGHT

Copyright 2017-2020 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>.

### 3.0.7 SSL\_CTX\_SET\_TLSEXT\_SERVERNAME\_CALLBACK(3oss)