

# sl\_mqtt\_client

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

## Modules

Here is a list of all modules:

- [SL MQTT Client API](#)
  - [SL MQTT Client Events](#)
  - [SL MQTT Oper Paramters](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

[Modules](#) | [Classes](#) | [Functions](#)

## SL MQTT Client API

## **Modules**

---

**SL MQTT Client Events**

---

**SL MQTT Oper Paramters**

## Classes

```
struct SIMqttClientCbs_t  
struct SIMqttWill_t  
struct SIMqttServer_t  
struct SIMqttClientLibCfg_t  
struct SIMqttClientCtxCfg_t
```

## Functions

```
_i32 sl_ExtLib_MqttClientInit (const SIMqttClientLibCfg_t *cfg)
_i32 sl_ExtLib_MqttClientExit ()
void * sl_ExtLib_MqttClientCtxCreate (const SIMqttClientCtxCfg_t
                                      *ctx_cfg, const SIMqttClientCbs_t *msg_cbs, void *app_hdl)
_i32 sl_ExtLib_MqttClientCtxDelete (void *cli_ctx)
_i32 sl_ExtLib_MqttClientSet (void *cli_ctx, _i32 param, const void
                             *value, _u32 len)
_i32 sl_ExtLib_MqttClientGet (void *cli_ctx, _i32 param, void
                             *value, _u32 len)
_i32 sl_ExtLib_MqttClientConnect (void *cli_ctx, bool clean, _u16
                                 keep_alive_time)
_i32 sl_ExtLib_MqttClientDisconnect (void *cli_ctx)
_i32 sl_ExtLib_MqttClientSub (void *cli_ctx, char *const *topics,
                             _u8 *qos, _i32 count)
_i32 sl_ExtLib_MqttClientUnsub (void *cli_ctx, char *const *topics,
                               _i32 count)
_i32 sl_ExtLib_MqttClientSend (void *cli_ctx, const char *topic,
                             const void *data, _i32 len, char qos, bool retain)
```

## Function Documentation

```
_i32 sl_ExtLib_MqttClientConnect( void * cli_ctx,
                                  bool clean,
                                  _u16 keep_alive_time
)
```

CONNECT to the server. This routine establishes a connection with the server for MQTT transactions. The caller should specify a time-period with-in which the implementation should send a message to the server to keep-alive the connection.

### Parameters:

- |                             |  |
|-----------------------------|--|
| [in] <b>cli_ctx</b>         | refers to the handle to the client context   |
| [in] <b>clean</b>           | assert to make a clean start and purge the previous session  |
| [in] <b>keep_alive_time</b> | the maximum time within which client should send a message to server.<br>The unit of the interval is in seconds. |

### Returns:

on success, variable header of CONNACK message in network byte order. Lowest Byte[Byte0] contains CONNACK Return Code. Byte1 Contains Session Present Bit. on failure returns(-1)

```
void* sl_ExtLib_MqttClientCtxCreate ( const SIMqttClientCtxCfg_t *
                                       const SIMqttClientCbs_t *
                                       void *
)
```

Create a new client context to connect to a server. A context has to be created prior to invoking the client services.

**Parameters:**

- [in] **ctx\_cfg** refers to client context configuration parameters
- [in] **msg\_cbs** refers to callbacks into application
- [in] **app** refers to the application callback to be returned on callback

### `_i32 sl_ExtLib_MqttClientCtxDelete( void * cli_ctx )`

Deletes the specified client context.

```
\param[in] cli_ctx refers to client context to be deleted  
\return Success (0) or Failure (< 0)
```

### `_i32 sl_ExtLib_MqttClientDisconnect( void * cli_ctx )`

DISCONNECT from the server. The caller must use this service to close the connection with the server.

**Parameters:**

- [in] **cli\_ctx** refers to the handle to the client context

**Returns:**

Success (0) or Failure (< 0)

### `_i32 sl_ExtLib_MqttClientExit( )`

Exit the SL MQTT Implementation.

```
\return Success (0) or Failure (-1)
```

### `_i32 sl_ExtLib_MqttClientGet( void * cli_ctx,`

```
    _i32 param,  
    void * value,  
    _u32 len  
)
```

## **\_i32 sl\_ExtLib\_MqttClientInit ( const SIMqttClientLibCfg\_t \* cfg )**

Initialize the SL MQTT Implementation. A caller must initialize the MQTT implementation prior to using its services.

### **Parameters:**

[in] **cfg** refers to client lib configuration parameters

### **Returns:**

Success (0) or Failure (-1)

```
_i32 sl_ExtLib_MqttClientSend ( void * cli_ctx,  
                                const char * topic,  
                                const void * data,  
                                _i32 len,  
                                char qos,  
                                bool retain  
)
```

PUBLISH a named message to the server. In addition to the PUBLISH specific parameters, the caller can indicate whether the routine should block until the time the message has been acknowledged by the server. This is applicable only for non-QoS0 messages.

In case, the app has chosen not to await for the ACK from the server, the SL MQTT implementation will notify the app about the subscription through the callback routine.

### Parameters:

- [in] **cli\_ctx** refers to the handle to the client context
- [in] **topic** topic of the data to be published. It is NULL terminated.
- [in] **data** binary data to be published
- [in] **len** length of the data
- [in] **qos** QoS for the publish message
- [in] **retain** assert if server should retain the message
- [in] **flags** Command flag. Refer to sl\_mqtt\_cl\_cmdflags

### Returns:

Success(transaction Message ID) or Failure(< 0)

```
_i32 sl_ExtLib_MqttClientSet( void *          cli_ctx,
                               _i32            param,
                               const void *    value,
                               _u32            len
                           )
```

Set parameters in SL MQTT implementation. The caller must configure these parameters prior to invoking any MQTT transaction.

### Note:

The implementation does not copy the contents referred. Therefore, the caller must ensure that contents are persistent in the memory.

### Parameters:

- [in] **cli\_ctx** refers to the handle to the client context
- [in] **param** identifies parameter to set. Refer to [SL MQTT Oper Paramters](#)
- [in] **value** refers to the place-holder of value to be set
- [in] **len** length of the value of the parameter

**Returns:**

Success (0) or Failure (-1)

```
_i32 sl_ExtLib_MqttClientSub ( void *           cli_ctx,
                                char *const * topics,
                                _u8 *          qos,
                                _i32           count
)
```

SUBSCRIBE a set of topics. To receive data about a set of topics from the server, the app through this routine must subscribe to those topic names with the server. The caller can indicate whether the routine should block until a time, the message has been acknowledged by the server.

In case, the app has chosen not to await for the ACK from the server, the SL MQTT implementation will notify the app about the subscription through the callback routine.

**Parameters:**

- [in]      **cli\_ctx** refers to the handle to the client context
- [in]      **topics** set of topic names to subscribe. It is an array of pointers to NUL terminated strings.
- [in, out] **qos** array of qos values for each topic in the same order of the topic array. If configured to await for SUB-ACK from server, the array will contain qos responses for topics from the server.
- [in]      **count** number of such topics

**Returns:**

Success(transaction Message ID) or Failure(< 0)

```
_i32 sl_ExtLib_MqttClientUnsub ( void *           cli_ctx,
                                char *const * topics,
                                _i32          count
)
```

UNSUBSCRIBE a set of topics. The app should use this service to stop receiving data for the named topics from the server. The caller can indicate whether the routine should block until a time, the message has been acknowledged by the server.

In case, the app has chosen not to await for the ACK from the server, the SL MQTT implementation will notify the app about the subscription through the callback routine.

#### Parameters:

- [in] **cli\_ctx** refers to the handle to the client context
- [in] **topics** set of topics to be unsubscribed. It is an array of pointers to NUL terminated strings.
- [in] **count** number of topics to be unsubscribed

#### Returns:

Success(transaction Message ID) or Failure(< 0)

# sl\_mqtt\_client

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

Defines

## SL MQTT Client Events

[SL MQTT Client API](#)

## Defines

```
#define SL_MQTT_CL_EVT_PUBACK 0x04
#define SL_MQTT_CL_EVT_PUBCOMP 0x07
#define SL_MQTT_CL_EVT_SUBACK 0x09
#define SL_MQTT_CL_EVT_UNSUBACK 0x0B
```

## Define Documentation

**#define SL\_MQTT\_CL\_EVT\_PUBACK 0x04**

PUBACK has been received from the server

**#define SL\_MQTT\_CL\_EVT\_PUBCOMP 0x07**

PUBCOMP has been received from the server

**#define SL\_MQTT\_CL\_EVT\_SUBACK 0x09**

SUBACK has been received from the server

**#define SL\_MQTT\_CL\_EVT\_UNSUBACK 0x0B**

UNSUBACK has been received from the server

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

Defines

## SL MQTT Oper Paramters

[SL MQTT Client API](#)

## Defines

```
#define SL_MQTT_PARAM_CLIENT_ID 0x01
#define SL_MQTT_PARAM_USER_NAME 0x02
#define SL_MQTT_PARAM_PASS_WORD 0x03
#define SL_MQTT_PARAM_TOPIC_QOS1 0x04
#define SL_MQTT_PARAM_WILL_PARAM 0x05
```

---

## Define Documentation

**#define SL\_MQTT\_PARAM\_CLIENT\_ID 0x01**

Refers to Client ID

**#define SL\_MQTT\_PARAM\_PASS\_WORD 0x03**

Pass-word of client

**#define SL\_MQTT\_PARAM\_TOPIC\_QOS1 0x04**

Set a QoS1 SUB topic

**#define SL\_MQTT\_PARAM\_USER\_NAME 0x02**

User name of client

**#define SL\_MQTT\_PARAM\_WILL\_PARAM 0x05**

Set a WILL topic,Will Message, Will QOS,Will Retain

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

Public Attributes

## SI~~M~~qttClientCbs\_t Struct Reference

[SL MQTT Client API](#)

#include <[sl\\_mqtt\\_client.h](#)>

List of all members.

## Public Attributes

```
void(* sl_ExtLib_MqttRecv )(void *app_hdl, const char *topstr, _i32  
void(* toplen, const void *payload, _i32 pay_len, bool dup, unsigned  
char qos, bool retain)  
void(* sl_ExtLib_MqttEvent )(void *app_hdl, _i32 evt, const void  
*buf, _u32 len)  
void(* sl_ExtLib_MqttDisconnect )(void *app_hdl)
```

## Detailed Description

Callbacks Routines The routines are invoked by SL Implementation onto Client application

**Note:**

The user applications implement the callbacks that are registered with the libraries. While using the MQTT library, invoking the core library APIs from a callback should be avoided and can lead to lockup scenarios. It is recommended to signal another task from the callback routines invoked from the library and invoke the core library API calls from that task.

---

## Member Data Documentation

**void(\* SIMqttClientCbs\_t::sl\_ExtLib\_MqttDisconnect)(void \*app\_hdl)**

Notifies the client app about the termination of MQTT connection.  
After servicing this callback, the client-app can destroy associated context if it no longer required

**Parameters:**

[in] **app\_hdl** application handle returned

**void(\* SIMqttClientCbs\_t::sl\_ExtLib\_MqttEvent)(void \*app\_hdl, \_i32**

Indication of event either from the server or implementation generated. These events are notified as part of the processing carried out by the internal recv task of the SL implementation. The application must populate the callback to receive events about the progress made by the SL Mqtt layer.

This handler is used by the SL Mqtt Layer to report acknowledgements from the server, in case, the application has chosen not to block the service invokes till the arrival of the corresponding ACK.

**Parameters:**

[in] **app\_hdl** application handle returned

[in] **evt** identifier to the reported event. Refer to [SL MQTT Client Events](#)

[in] **buf** points to buffer

[in] **len** length of buffer

**Note:**

**void(\* SIMqttClientCbs\_t::sl\_ExtLib\_MqttRecv)(void \*app\_hdl, const char \*topic, const void \*payload, size\_t payload\_len, size\_t toplen, bool dup, int qos, bool retain)**

Callback routine to receive a PUBLISH from the server. The client app must provide this routine for the instances where it has subscribed to certain set of topics from the server. The callback is invoked in the context of the internal SL Receive Task.

**Parameters:**

- [in] **app\_hdl** application handle returned
- [in] **topstr** name of topic published by the server. Not NUL terminated.
- [in] **toplen** length of the topic name published by the server.
- [in] **payload** refers to payload published by the server.
- [in] **pay\_len** length of the payload.
- [in] **dup** assert to indicate that it is re-send by the server
- [in] **qoS** quality of service of the published message
- [in] **retain** asserted to indicate that a retained message has been published

**Returns:**

none.

The documentation for this struct was generated from the following file:

- D:/Project/SimpleLink/mqtt/doxygen/client/**sl\_mqtt\_client.h**

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by

**doxygen** 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	Public Attributes

## SI~~M~~qttWill\_t Struct Reference

[SL MQTT Client API](#)

```
#include <sl_mqtt_client.h>
```

List of all members.

## Public Attributes

```
const char * will_topic  
const char * will_msg  
char will_qos  
bool retain
```

## Member Data Documentation

**bool SIMqttWill\_t::retain**

Retain Flag

**const char\* SIMqttWill\_t::will\_msg**

Will message

**char SIMqttWill\_t::will\_qos**

Will Qos

**const char\* SIMqttWill\_t::will\_topic**

Will Topic

---

The documentation for this struct was generated from the following file:

- D:/Project/SimpleLink/mqtt/doxygen/client/[sl\\_mqtt\\_client.h](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

Public Attributes

## SI\_MqttServer\_t Struct Reference

[SL MQTT Client API](#)

```
#include <sl_mqtt_client.h>
```

List of all members.

## Public Attributes

```
_u32 netconn_flags  
const char * server_addr  
_u16 port_number  
char method  
_u32 cipher  
_u32 n_files  
char *const * secure_files
```

---

## **Detailed Description**

Secure Socket Parameters to open a secure connection

---

## Member Data Documentation

`_u32 SIMqttServer_t::cipher`

Cipher to tcp secured socket

`char SIMqttServer_t::method`

Method to tcp secured socket

`_u32 SIMqttServer_t::n_files`

Number of files for secure transfer

`_u32 SIMqttServer_t::netconn_flags`

Enumerate connection type

`_u16 SIMqttServer_t::port_number`

Port number of MQTT server

`char* const* SIMqttServer_t::secure_files`

`const char* SIMqttServer_t::server_addr`

Server Address: URL or IP

---

The documentation for this struct was generated from the following file:

- D:/Project/SimpleLink/mqtt/doxygen/client/[\*\*sl\\_mqtt\\_client.h\*\*](#)
- 

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files	
Class List	Class Index	Class Members		Public Attributes

## SI~~M~~qttClientLibCfg\_t Struct Reference

[SL MQTT Client API](#)

```
#include <sl_mqtt_client.h>
```

List of all members.

## Public Attributes

```
_u16 loopback_port  
_u32 rx_tsk_priority  
_u32 resp_time  
bool aux_debug_en  
_i32(* dbg_print )(const char *pcFormat,...)
```

## **Detailed Description**

MQTT Lib structure which holds Initialization Data

---

## Member Data Documentation

**bool SIMqttClientLibCfg\_t::aux\_debug\_en**

Assert to indicate additional debug info

**\_i32(\* SIMqttClientLibCfg\_t::dbg\_print)(const char \*pcFormat,...)**

Print debug information

**\_u16 SIMqttClientLibCfg\_t::loopback\_port**

< Loopback port is used to manage lib internal functioning in case of connections to multiple servers simultaneously is desired. Loopback port = 0, implies connection to only single server Loopback port != 0, implies connection to multiple servers

**\_u32 SIMqttClientLibCfg\_t::resp\_time**

Reasonable response time (seconds) from server

**\_u32 SIMqttClientLibCfg\_t::rx\_tsk\_priority**

Priority of the receive task

The documentation for this struct was generated from the following file:

- D:/Project/SimpleLink/mqtt/doxygen/client/**sl\_mqtt\_client.h**

doxygen 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files	
Class List	Class Index	Class Members		Public Attributes

## SI~~M~~qttClientCtxCfg\_t Struct Reference

[SL MQTT Client API](#)

```
#include <sl_mqtt_client.h>
```

List of all members.

## Public Attributes

<code>SIMqttServer_t server_info</code>
<code>bool mqtt_mode31</code>
<code>bool blocking_send</code>

---

## **Detailed Description**

MQTT client context configuration structure

---

## Member Data Documentation

**bool SIMqttClientCtxCfg\_t::blocking\_send**

Select the mode of operation for send APIs (PUB, SUB, UNSUB).  
false - callback, true - blocking

**bool SIMqttClientCtxCfg\_t::mqtt\_mode31**

Operate LIB in MQTT 3.1 mode; default is 3.1.1. false - default( 3.1.1)  
& true - 3.1)

**SIMqttServer\_t SIMqttClientCtxCfg\_t::server\_info**

Server information

The documentation for this struct was generated from the following file:

- D:/Project/SimpleLink/mqtt/doxygen/client/**sl\_mqtt\_client.h**

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
**doxygen** 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

[SIMqttClientCbs\\_t](#)

[SIMqttClientCtxCfg\\_t](#)

[SIMqttClientLibCfg\\_t](#)

[SIMqttServer\\_t](#)

[SIMqttWill\\_t](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## Class Index

S

S

[\*\*SIMqttClientCtxCfg\\_t\*\*](#)    [\*\*SIMqttServer\\_t\*\*](#)  
[\*\*SIMqttClientLibCfg\\_t\*\*](#)    [\*\*SIMqttWill\\_t\*\*](#)

[\*\*SIMqttClientCbs\\_t\*\*](#)

S

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[\*\*doxygen\*\*](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	
All	Variables		

Here is a list of all class members with links to the classes they belong to:

- aux\_debug\_en : [SIMqttClientLibCfg\\_t](#)
- blocking\_send : [SIMqttClientCtxCfg\\_t](#)
- cipher : [SIMqttServer\\_t](#)
- dbg\_print : [SIMqttClientLibCfg\\_t](#)
- loopback\_port : [SIMqttClientLibCfg\\_t](#)
- method : [SIMqttServer\\_t](#)
- mqtt\_mode31 : [SIMqttClientCtxCfg\\_t](#)
- n\_files : [SIMqttServer\\_t](#)
- netconn\_flags : [SIMqttServer\\_t](#)
- port\_number : [SIMqttServer\\_t](#)
- resp\_time : [SIMqttClientLibCfg\\_t](#)
- retain : [SIMqttWill\\_t](#)
- rx\_tsk\_priority : [SIMqttClientLibCfg\\_t](#)
- secure\_files : [SIMqttServer\\_t](#)
- server\_addr : [SIMqttServer\\_t](#)
- server\_info : [SIMqttClientCtxCfg\\_t](#)
- sl\_ExtLib\_MqttDisconnect : [SIMqttClientCbs\\_t](#)
- sl\_ExtLib\_MqttEvent : [SIMqttClientCbs\\_t](#)
- sl\_ExtLib\_MqttRecv : [SIMqttClientCbs\\_t](#)
- will\_msg : [SIMqttWill\\_t](#)
- will\_qos : [SIMqttWill\\_t](#)
- will\_topic : [SIMqttWill\\_t](#)

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	
All	Variables		

- aux\_debug\_en : [SIMqttClientLibCfg\\_t](#)
- blocking\_send : [SIMqttClientCtxCfg\\_t](#)
- cipher : [SIMqttServer\\_t](#)
- dbg\_print : [SIMqttClientLibCfg\\_t](#)
- loopback\_port : [SIMqttClientLibCfg\\_t](#)
- method : [SIMqttServer\\_t](#)
- mqtt\_mode31 : [SIMqttClientCtxCfg\\_t](#)
- n\_files : [SIMqttServer\\_t](#)
- netconn\_flags : [SIMqttServer\\_t](#)
- port\_number : [SIMqttServer\\_t](#)
- resp\_time : [SIMqttClientLibCfg\\_t](#)
- retain : [SIMqttWill\\_t](#)
- rx\_tsk\_priority : [SIMqttClientLibCfg\\_t](#)
- secure\_files : [SIMqttServer\\_t](#)
- server\_addr : [SIMqttServer\\_t](#)
- server\_info : [SIMqttClientCtxCfg\\_t](#)
- sl\_ExtLib\_MqttDisconnect : [SIMqttClientCbs\\_t](#)
- sl\_ExtLib\_MqttEvent : [SIMqttClientCbs\\_t](#)
- sl\_ExtLib\_MqttRecv : [SIMqttClientCbs\\_t](#)
- will\_msg : [SIMqttWill\\_t](#)
- will\_qos : [SIMqttWill\\_t](#)
- will\_topic : [SIMqttWill\\_t](#)

# sl\_mqtt\_client

Main Page

Modules

Classes

Files

File List

File Members

## File List

Here is a list of all files with brief descriptions:

[D:/Project/SimpleLink/mqtt/doxygen/client/sl\\_mqtt\\_client.h \[code\]](#)

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

[Main Page](#)[Modules](#)[Classes](#)[Files](#)[File List](#)[File Members](#)[Classes](#) | [Defines](#) | [Functions](#)

**D:/Project/SimpleLink/mqtt/doxygen/client/sl\_m**

## File Reference

```
#include <stdio.h> #include <string.h>
#include <stdbool.h>
#include "simplelink.h"
```

[Go to the source code of this file.](#)

## Classes

```
struct SIMqttClientCbs_t  
struct SIMqttWill_t  
struct SIMqttServer_t  
struct SIMqttClientLibCfg_t  
struct SIMqttClientCtxCfg_t
```

## Defines

```
#define SL_MQTT_CL_EVT_PUBACK 0x04
#define SL_MQTT_CL_EVT_PUBCOMP 0x07
#define SL_MQTT_CL_EVT_SUBACK 0x09
#define SL_MQTT_CL_EVT_UNSUBACK 0x0B
#define SL_MQTT_NETCONN_IP6 0x04
#define SL_MQTT_NETCONN_URL 0x08
#define SL_MQTT_NETCONN_SEC 0x10
#define SL_MQTT_PARAM_CLIENT_ID 0x01
#define SL_MQTT_PARAM_USER_NAME 0x02
#define SL_MQTT_PARAM_PASS_WORD 0x03
#define SL_MQTT_PARAM_TOPIC_QOS1 0x04
#define SL_MQTT_PARAM_WILL_PARAM 0x05
```

## Functions

```
_i32 sl_ExtLib_MqttClientInit (const SIMqttClientLibCfg_t *cfg)
_i32 sl_ExtLib_MqttClientExit ()
void * sl_ExtLib_MqttClientCtxCreate (const SIMqttClientCtxCfg_t
                                      *ctx_cfg, const SIMqttClientCbs_t *msg_cbs, void *app_hdl)
_i32 sl_ExtLib_MqttClientCtxDelete (void *cli_ctx)
_i32 sl_ExtLib_MqttClientSet (void *cli_ctx, _i32 param, const void
                             *value, _u32 len)
_i32 sl_ExtLib_MqttClientGet (void *cli_ctx, _i32 param, void
                             *value, _u32 len)
_i32 sl_ExtLib_MqttClientConnect (void *cli_ctx, bool clean, _u16
                                 keep_alive_time)
_i32 sl_ExtLib_MqttClientDisconnect (void *cli_ctx)
_i32 sl_ExtLib_MqttClientSub (void *cli_ctx, char *const *topics,
                             _u8 *qos, _i32 count)
_i32 sl_ExtLib_MqttClientUnsub (void *cli_ctx, char *const *topics,
                               _i32 count)
_i32 sl_ExtLib_MqttClientSend (void *cli_ctx, const char *topic,
                             const void *data, _i32 len, char qos, bool retain)
```

## Define Documentation

**#define SL\_MQTT\_NETCONN\_IP6 0x04**

Assert for IPv6 connection, otherwise IPv4

**#define SL\_MQTT\_NETCONN\_SEC 0x10**

Connection to server must be secure (TLS)

**#define SL\_MQTT\_NETCONN\_URL 0x08**

Server address is an URL and not IP address

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

Here is a list of all file members with links to the files they belong to:

- sl\_ExtLib\_MqttClientConnect() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientCtxCreate() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientCtxDelete() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientDisconnect() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientExit() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientGet() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientInit() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSend() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSet() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSub() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientUnsub() : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_PUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_PUBCOMP : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_SUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_UNSUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_IP6 : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_SEC : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_URL : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_CLIENT\_ID : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_PASS\_WORD : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_TOPIC\_QOS1 : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_USER\_NAME : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_WILL\_PARAM : [sl\\_mqtt\\_client.h](#)

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

- sl\_ExtLib\_MqttClientConnect() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientCtxCreate() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientCtxDelete() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientDisconnect() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientExit() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientGet() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientInit() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSend() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSet() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientSub() : [sl\\_mqtt\\_client.h](#)
- sl\_ExtLib\_MqttClientUnsub() : [sl\\_mqtt\\_client.h](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

- SL\_MQTT\_CL\_EVT\_PUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_PUBCOMP : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_SUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_CL\_EVT\_UNSUBACK : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_IP6 : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_SEC : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_NETCONN\_URL : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_CLIENT\_ID : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_PASS\_WORD : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_TOPIC\_QOS1 : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_USER\_NAME : [sl\\_mqtt\\_client.h](#)
- SL\_MQTT\_PARAM\_WILL\_PARAM : [sl\\_mqtt\\_client.h](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
File List	File Members		

## D:/Project/SimpleLink/mqtt/doxygen/client/sl\_mqtt\_client.h

Go to the documentation of this file.

```
00001 /*****  
*****  
00002 Copyright (c) (2014) Texas Instruments Incorporated  
00003 All rights reserved not granted herein.  
00004  
00005 Limited License.  
00006  
00007 Texas Instruments Incorporated grants a world-wide,  
00008 royalty-free, non-exclusive  
00009 license under copyrights and patents it now  
00010 or hereafter owns or controls to make,  
00011 have made, use, import, offer to sell and sell ("Utilize") this software subject  
00012 to the terms herein. With respect to the foregoing patent license, such license  
00013 is granted solely to the extent that any such patent is necessary to Utilize the  
00014 software alone. The patent license shall not apply to any combinations which  
00015 include this software, other than combinations with devices manufactured by or  
00016 for TI (TI Devices). No hardware patent is licensed hereunder.  
00017  
00018 Redistributions must preserve existing copyright notices and reproduce this license  
00019  
00020
```

00017 (including the above copyright notice and th  
e disclaimer and (if applicable) source  
00018 code license limitations below) in the docum  
entation and/or other materials provided  
00019 with the distribution  
00020  
00021 Redistribution and use in binary form, witho  
ut modification, are permitted provided  
00022 that the following conditions are met:  
00023 \* No reverse engineering, decompilation, or  
disassembly of this software is  
00024 permitted with respect to any software pro  
vided in binary form.  
00025 \* any redistribution and use are licensed by  
TI for use only with TI Devices.  
00026 \* Nothing shall obligate TI to provide you w  
ith source code for the software  
00027 licensed and provided to you in object cod  
e.  
00028  
00029 If software source code is provided to you,  
modification and redistribution of the  
00030 source code are permitted provided that the  
following conditions are met;  
00031 \* any redistribution and use of the source c  
ode, including any resulting derivative  
00032 works, are licensed by TI for use only wit  
h TI Devices.  
00033 \* any redistribution and use of any object c  
ode compiled from the source code and  
00034 any resulting derivative works, are licens  
ed by TI for use only with TI Devices.  
00035  
00036 Neither the name of Texas Instruments Incorp  
orated nor the names of its suppliers  
00037 may be used to endorse or promote products d  
erived from this software without

00038 specific prior written permission.

00039

00040 DISCLAIMER.

00041

00042 THIS SOFTWARE IS PROVIDED BY TI AND TI'S LIC  
ENSORS "AS IS" AND ANY EXPRESS OR IMPLIED

00043 WARRANTIES, INCLUDING, BUT NOT LIMITED TO, T  
HE IMPLIED WARRANTIES OF MERCHANTABILITY

00044 AND FITNESS FOR A PARTICULAR PURPOSE ARE DIS  
CLAIMED. IN NO EVENT SHALL TI AND TI'S

00045 LICENSORS BE LIABLE FOR ANY DIRECT, INDIRECT  
, INCIDENTAL, SPECIAL, EXEMPLARY, OR

00046 CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LI  
MITED TO, PROCUREMENT OF SUBSTITUTE

00047 GOODS OR SERVICES; LOSS OF USE, DATA, OR PRO  
FITS; OR BUSINESS INTERRUPTION)

00048 HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY  
, WHETHER IN CONTRACT, STRICT LIABILITY,

00049 OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
ARISING IN ANY WAY OUT OF THE USE OF

00050 THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIB  
ILITY OF SUCH DAMAGE.

00051

00052 \*\*\*\*\*/

00053

00054 #include <stdio.h>

00055 #include <string.h>

00056 #include <stdbool.h>

00057 #include "simplelink.h"

00058

00059 #ifndef \_\_SL\_MQTT\_H\_\_

00060 #define \_\_SL\_MQTT\_H\_\_

00061

00062 #ifdef \_\_cplusplus

00063 extern "C"

00064 {

```
00065 #endif
00066
00134 #define SL_MQTT_CL_EVT_PUBACK    0x04
00135 #define SL_MQTT_CL_EVT_PUBCOMP   0x07
00136 #define SL_MQTT_CL_EVT_SUBACK    0x09
00137 #define SL_MQTT_CL_EVT_UNSUBACK  0x0B /* End Client events */
00139
00140
00141     /* Define server structure which holds
   , server address and port number.
00142     These values are set by the sl_MqttSet
   API and retrieved by sl_MqttGet API*/
00143
00144     typedef struct {
00155
00171             void (*sl_ExtLib_MqttRecv)(v
oid *app_hdl, const char *topstr, _i32 toplen,
00172
const void *payload, _i32 pay_len,
00173
bool dup, unsigned char qos,
00174
bool retain);
00175
00192             void (*sl_ExtLib_MqttEvent)(
void *app_hdl, _i32 evt, const void *buf,
00193
_u32 len);
00194
00201             void (*sl_ExtLib_MqttDisconn
)(void *app_hdl);
00202
00203
00204     } SlMqttClientCbs_t;
00205
00206     typedef struct {
```

```
00207
00208         const char           *will_topic
00209 ;
00210         const char           *will_msg;
00211         char      will_qos;
00212         bool        retain;
00213     } SlMqttWill_t;
00214
00215
00216     typedef struct {
00217
00218 #define SL_MQTT_NETCONN_IP6    0x04
00219 #define SL_MQTT_NETCONN_URL   0x08
00220 #define SL_MQTT_NETCONN_SEC   0x10
00221
00222         _u32      netconn_flags;
00223         const char *server_addr;
00224
00225         _u16      port_number;
00226         char      method;
00227         _u32      cipher;
00228         _u32      n_files;
00229         char * const *secure_
00230         files; /* SL needs 4 files*/
00231
00232     } SlMqttServer_t;
00233
00234
00235     typedef struct
00236     {
00237
00238         _u16  loopback_port;
00239         _u32  rx_tsk_priority;
00240         _u32  resp_time;
00241         bool aux_debug_en;
00242
00243
00244         _i32 (*dbg_print)(const char
```

```
    *pcFormat, ...);  
00246          } SlMqttClientLibCfg_t;  
00247  
00248      typedef struct  
00250      {  
00252          SlMqttServer_t server_info;  
00253          bool mqtt_mode31;  
00254          bool blocking_send;  
00255      };  
00256      } SlMqttClientCtxCfg_t;  
00257  
00264      _i32 sl_ExtLib_MqttClientInit(const  
SlMqttClientLibCfg_t *cfg);  
00265  
00270      _i32 sl_ExtLib_MqttClientExit();  
00271  
00279      void *sl_ExtLib_MqttClientCtxCreate(  
const SlMqttClientCtxCfg_t *ctx_cfg,  
00280  
    const SlMqttClientCbs_t *msg_cbs,  
00281  
    void *app_hdl);  
00282  
00288      _i32 sl_ExtLib_MqttClientCtxDelete(v  
oid *cli_ctx);  
00289  
00293 #define SL_MQTT_PARAM_CLIENT_ID 0x01  
00294 #define SL_MQTT_PARAM_USER_NAME 0x02  
00295 #define SL_MQTT_PARAM_PASS_WORD 0x03  
00296 #define SL_MQTT_PARAM_TOPIC_QOS1 0x04  
00297 #define SL_MQTT_PARAM_WILL_PARAM 0x05  
00314      _i32 sl_ExtLib_MqttClientSet(void *c  
li_ctx, _i32 param, const void *value, _u32 len);  
00315
```

```
00316         /*\brief None defined at the moment
00317         */
00318         _i32 sl_ExtLib_MqttClientGet(void *c
li_ctx, _i32 param, void *value, _u32 len);
00319
00320
00335         _i32 sl_ExtLib_MqttClientConnect(void
*cli_ctx, bool clean, _u16 keep_alive_time);
00336
00345         _i32 sl_ExtLib_MqttClientDisconnect(
void *cli_ctx);
00346
00347
00368         _i32 sl_ExtLib_MqttClientSub(void *c
li_ctx, char* const *topics,
00369                                         _u8 *qos
, _i32 count);
00370
00371
00389         _i32 sl_ExtLib_MqttClientUnsub(void
*cli_ctx, char* const *topics, _i32 count);
00390
00391
00410         _i32 sl_ExtLib_MqttClientSend(void *
cli_ctx, const char *topic,
00411                                         const
void *data, _i32 len,
00412                                         char q
os, bool retain);
00413
00416         static inline _i32 sl_ExtLib_MqttCli
entPub(void *cli_ctx, const char *topic,
00417                                         const
void *data, _i32 len,
00418                                         char q
os, bool retain)
00419         {
```

```
00420                     return sl_ExtLib_MqttClients
end(cli_ctx, topic, data, len,
00421                                         qo
s, retain);
00422         }
00423 /* End Client API */
00425
00426 #ifdef __cplusplus
00427 }
00428 #endif
00429
00430
00431
00432 #endif // __SL_MQTT_H__
```

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## SIMqttClientCbs\_t Member List

This is the complete list of members for **SIMqttClientCbs\_t**, including all inherited members.

**sl\_ExtLib\_MqttDisconnect** **SIMqttClientCbs\_t**

**sl\_ExtLib\_MqttEvent** **SIMqttClientCbs\_t**

**sl\_ExtLib\_MqttRecv** **SIMqttClientCbs\_t**

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
**[doxygen](#)** 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## SIMqttWill\_t Member List

This is the complete list of members for [SIMqttWill\\_t](#), including all inherited members.

retain	<a href="#">SIMqttWill_t</a>
will_msg	<a href="#">SIMqttWill_t</a>
will_qos	<a href="#">SIMqttWill_t</a>
will_topic	<a href="#">SIMqttWill_t</a>

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## SIMqttServer\_t Member List

This is the complete list of members for [SIMqttServer\\_t](#), including all inherited members.

cipher	<a href="#">SIMqttServer_t</a>
method	<a href="#">SIMqttServer_t</a>
n_files	<a href="#">SIMqttServer_t</a>
netconn_flags	<a href="#">SIMqttServer_t</a>
port_number	<a href="#">SIMqttServer_t</a>
secure_files	<a href="#">SIMqttServer_t</a>
server_addr	<a href="#">SIMqttServer_t</a>

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## SIMqttClientLibCfg\_t Member List

This is the complete list of members for [SIMqttClientLibCfg\\_t](#), including all inherited members.

<code>aux_debug_en</code>	<a href="#">SIMqttClientLibCfg_t</a>
<code>dbg_print</code>	<a href="#">SIMqttClientLibCfg_t</a>
<code>loopback_port</code>	<a href="#">SIMqttClientLibCfg_t</a>
<code>resp_time</code>	<a href="#">SIMqttClientLibCfg_t</a>
<code>rx_tsk_priority</code>	<a href="#">SIMqttClientLibCfg_t</a>

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0

# sl\_mqtt\_client

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

## **SIMqttClientCtxCfg\_t Member List**

This is the complete list of members for [\*\*SIMqttClientCtxCfg\\_t\*\*](#), including all inherited members.

[\*\*blocking\\_send\*\*](#) [\*\*SIMqttClientCtxCfg\\_t\*\*](#)

[\*\*mqtt\\_mode31\*\*](#) [\*\*SIMqttClientCtxCfg\\_t\*\*](#)

[\*\*server\\_info\*\*](#) [\*\*SIMqttClientCtxCfg\\_t\*\*](#)

---

Generated on Thu Jan 15 2015 18:26:27 for sl\_mqtt\_client by  
[doxygen](#) 1.8.0