

FreeCoAP

Generated by Doxygen 1.8.9.1

Wed Jan 27 2016 20:34:23



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	coap_client_t Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	num_retrans . . . . .	5
3.1.2.2	sd . . . . .	5
3.1.2.3	server_host . . . . .	5
3.1.2.4	server_port . . . . .	6
3.1.2.5	server_sin . . . . .	6
3.1.2.6	server_sin_len . . . . .	6
3.1.2.7	timeout . . . . .	6
3.1.2.8	timer_fd . . . . .	6
3.2	coap_msg_op Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Data Documentation . . . . .	7
3.2.2.1	len . . . . .	7
3.2.2.2	next . . . . .	7
3.2.2.3	num . . . . .	7
3.2.2.4	val . . . . .	7
3.3	coap_msg_op_list_t Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Member Data Documentation . . . . .	8
3.3.2.1	first . . . . .	8
3.3.2.2	last . . . . .	8
3.4	coap_msg_t Struct Reference . . . . .	8
3.4.1	Detailed Description . . . . .	9

3.4.2	Member Data Documentation . . . . .	9
3.4.2.1	code_class . . . . .	9
3.4.2.2	code_detail . . . . .	9
3.4.2.3	msg_id . . . . .	9
3.4.2.4	op_list . . . . .	9
3.4.2.5	payload . . . . .	9
3.4.2.6	payload_len . . . . .	9
3.4.2.7	token . . . . .	9
3.4.2.8	token_len . . . . .	9
3.4.2.9	type . . . . .	9
3.4.2.10	ver . . . . .	9
3.5	coap_server Struct Reference . . . . .	10
3.5.1	Detailed Description . . . . .	10
3.5.2	Member Data Documentation . . . . .	10
3.5.2.1	handle . . . . .	10
3.5.2.2	msg_id . . . . .	11
3.5.2.3	sd . . . . .	11
3.5.2.4	sep_list . . . . .	11
3.5.2.5	trans . . . . .	11
3.6	coap_server_path Struct Reference . . . . .	11
3.6.1	Detailed Description . . . . .	11
3.6.2	Member Data Documentation . . . . .	11
3.6.2.1	next . . . . .	11
3.6.2.2	str . . . . .	12
3.7	coap_server_path_list_t Struct Reference . . . . .	12
3.7.1	Detailed Description . . . . .	12
3.7.2	Member Data Documentation . . . . .	12
3.7.2.1	first . . . . .	12
3.7.2.2	last . . . . .	12
3.8	coap_server_trans Struct Reference . . . . .	13
3.8.1	Detailed Description . . . . .	14
3.8.2	Member Data Documentation . . . . .	14
3.8.2.1	active . . . . .	14
3.8.2.2	client_addr . . . . .	14
3.8.2.3	client_sin . . . . .	14
3.8.2.4	client_sin_len . . . . .	14
3.8.2.5	last_use . . . . .	14
3.8.2.6	num_retrans . . . . .	14
3.8.2.7	req . . . . .	14
3.8.2.8	resp . . . . .	14

3.8.2.9	server	14
3.8.2.10	timeout	14
3.8.2.11	timer_fd	15
<b>4</b>	<b>File Documentation</b>	<b>17</b>
4.1	lib/include/coap_client.h File Reference	17
4.1.1	Detailed Description	18
4.1.2	Macro Definition Documentation	18
4.1.2.1	COAP_CLIENT_HOST_BUF_LEN	18
4.1.2.2	COAP_CLIENT_PORT_BUF_LEN	18
4.1.3	Function Documentation	19
4.1.3.1	coap_client_create	19
4.1.3.2	coap_client_destroy	20
4.1.3.3	coap_client_exchange	20
4.2	lib/include/coap_log.h File Reference	20
4.2.1	Detailed Description	21
4.2.2	Macro Definition Documentation	21
4.2.2.1	COAP_LOG_DEF_LEVEL	21
4.2.3	Enumeration Type Documentation	22
4.2.3.1	coap_log_level_t	22
4.2.4	Function Documentation	22
4.2.4.1	coap_log_debug	22
4.2.4.2	coap_log_error	22
4.2.4.3	coap_log_get_level	22
4.2.4.4	coap_log_info	22
4.2.4.5	coap_log_notice	22
4.2.4.6	coap_log_set_level	23
4.2.4.7	coap_log_warn	23
4.3	lib/include/coap_msg.h File Reference	23
4.3.1	Detailed Description	26
4.3.2	Macro Definition Documentation	26
4.3.2.1	coap_msg_get_code_class	26
4.3.2.2	coap_msg_get_code_detail	26
4.3.2.3	coap_msg_get_first_op	26
4.3.2.4	coap_msg_get_msg_id	27
4.3.2.5	coap_msg_get_payload	27
4.3.2.6	coap_msg_get_payload_len	27
4.3.2.7	coap_msg_get_token	27
4.3.2.8	coap_msg_get_token_len	27
4.3.2.9	coap_msg_get_type	27

4.3.2.10	<code>coap_msg_get_ver</code>	27
4.3.2.11	<code>coap_msg_is_empty</code>	27
4.3.2.12	<code>COAP_MSG_MAX_BUF_LEN</code>	27
4.3.2.13	<code>COAP_MSG_MAX_CODE_CLASS</code>	27
4.3.2.14	<code>COAP_MSG_MAX_CODE_DETAIL</code>	27
4.3.2.15	<code>COAP_MSG_MAX_MSG_ID</code>	27
4.3.2.16	<code>COAP_MSG_MAX_TOKEN_LEN</code>	28
4.3.2.17	<code>coap_msg_op_get_len</code>	28
4.3.2.18	<code>coap_msg_op_get_next</code>	28
4.3.2.19	<code>coap_msg_op_get_num</code>	28
4.3.2.20	<code>coap_msg_op_get_val</code>	28
4.3.2.21	<code>coap_msg_op_num_is_critical</code>	28
4.3.2.22	<code>coap_msg_op_num_is_unsafe</code>	28
4.3.2.23	<code>coap_msg_op_num_no_cache_key</code>	28
4.3.2.24	<code>coap_msg_op_set_len</code>	28
4.3.2.25	<code>coap_msg_op_set_next</code>	28
4.3.2.26	<code>coap_msg_op_set_num</code>	28
4.3.2.27	<code>coap_msg_op_set_val</code>	28
4.3.2.28	<code>COAP_MSG_OP_URI_PATH_MAX_LEN</code>	29
4.3.2.29	<code>COAP_MSG_OP_URI_PATH_NUM</code>	29
4.3.2.30	<code>COAP_MSG_VER</code>	29
4.3.3	Function Documentation	29
4.3.3.1	<code>coap_msg_add_op</code>	29
4.3.3.2	<code>coap_msg_check_critical_ops</code>	29
4.3.3.3	<code>coap_msg_check_unsafe_ops</code>	29
4.3.3.4	<code>coap_msg_copy</code>	30
4.3.3.5	<code>coap_msg_create</code>	30
4.3.3.6	<code>coap_msg_destroy</code>	30
4.3.3.7	<code>coap_msg_format</code>	30
4.3.3.8	<code>coap_msg_gen_rand_str</code>	31
4.3.3.9	<code>coap_msg_op_num_is_recognized</code>	31
4.3.3.10	<code>coap_msg_parse</code>	31
4.3.3.11	<code>coap_msg_parse_type_msg_id</code>	32
4.3.3.12	<code>coap_msg_reset</code>	32
4.3.3.13	<code>coap_msg_set_code</code>	32
4.3.3.14	<code>coap_msg_set_msg_id</code>	33
4.3.3.15	<code>coap_msg_set_payload</code>	33
4.3.3.16	<code>coap_msg_set_token</code>	33
4.3.3.17	<code>coap_msg_set_type</code>	34
4.4	<code>lib/include/coap_server.h</code> File Reference	34

4.4.1	Detailed Description	36
4.4.2	Macro Definition Documentation	36
4.4.2.1	COAP_SERVER_ADDR_BUF_LEN	36
4.4.2.2	COAP_SERVER_DIAG_PAYLOAD_LEN	36
4.4.2.3	COAP_SERVER_NUM_TRANS	36
4.4.3	Function Documentation	36
4.4.3.1	coap_server_add_sep_resp_uri_path	36
4.4.3.2	coap_server_create	37
4.4.3.3	coap_server_destroy	37
4.4.3.4	coap_server_get_next_msg_id	37
4.4.3.5	coap_server_run	37
4.5	lib/src/coap_client.c File Reference	38
4.5.1	Detailed Description	39
4.5.2	Macro Definition Documentation	39
4.5.2.1	COAP_CLIENT_ACK_TIMEOUT_SEC	39
4.5.2.2	COAP_CLIENT_MAX_RETRANSMIT	39
4.5.2.3	COAP_CLIENT_RESP_TIMEOUT_SEC	39
4.5.3	Function Documentation	39
4.5.3.1	coap_client_create	39
4.5.3.2	coap_client_destroy	39
4.5.3.3	coap_client_exchange	39
4.6	lib/src/coap_log.c File Reference	40
4.6.1	Detailed Description	41
4.6.2	Function Documentation	41
4.6.2.1	coap_log_debug	41
4.6.2.2	coap_log_error	41
4.6.2.3	coap_log_get_level	41
4.6.2.4	coap_log_info	41
4.6.2.5	coap_log_notice	41
4.6.2.6	coap_log_set_level	42
4.6.2.7	coap_log_warn	42
4.7	lib/src/coap_msg.c File Reference	42
4.7.1	Detailed Description	43
4.7.2	Macro Definition Documentation	43
4.7.2.1	coap_msg_op_list_get_first	43
4.7.2.2	coap_msg_op_list_get_last	43
4.7.2.3	coap_msg_op_list_is_empty	44
4.7.3	Function Documentation	44
4.7.3.1	coap_msg_add_op	44
4.7.3.2	coap_msg_check_critical_ops	44

4.7.3.3	<code>coap_msg_check_unsafe_ops</code>	44
4.7.3.4	<code>coap_msg_copy</code>	45
4.7.3.5	<code>coap_msg_create</code>	45
4.7.3.6	<code>coap_msg_destroy</code>	45
4.7.3.7	<code>coap_msg_format</code>	45
4.7.3.8	<code>coap_msg_gen_rand_str</code>	46
4.7.3.9	<code>coap_msg_op_num_is_recognized</code>	46
4.7.3.10	<code>coap_msg_parse</code>	46
4.7.3.11	<code>coap_msg_parse_type_msg_id</code>	46
4.7.3.12	<code>coap_msg_reset</code>	47
4.7.3.13	<code>coap_msg_set_code</code>	47
4.7.3.14	<code>coap_msg_set_msg_id</code>	47
4.7.3.15	<code>coap_msg_set_payload</code>	48
4.7.3.16	<code>coap_msg_set_token</code>	48
4.7.3.17	<code>coap_msg_set_type</code>	48
4.8	lib/src/coap_server.c File Reference	49
4.8.1	Detailed Description	50
4.8.2	Macro Definition Documentation	50
4.8.2.1	<code>COAP_SERVER_ACK_TIMEOUT_SEC</code>	50
4.8.2.2	<code>COAP_SERVER_MAX_RETRANSMIT</code>	50
4.8.3	Function Documentation	50
4.8.3.1	<code>coap_server_add_sep_resp_uri_path</code>	50
4.8.3.2	<code>coap_server_create</code>	50
4.8.3.3	<code>coap_server_destroy</code>	51
4.8.3.4	<code>coap_server_get_next_msg_id</code>	51
4.8.3.5	<code>coap_server_run</code>	51
	<b>Index</b>	<b>53</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">coap_client_t</a>		
Client structure	. . . . .	5
<a href="#">coap_msg_op</a>		
Option structure	. . . . .	6
<a href="#">coap_msg_op_list_t</a>		
Option linked-list structure	. . . . .	7
<a href="#">coap_msg_t</a>		
Message structure	. . . . .	8
<a href="#">coap_server</a>		
Server structure	. . . . .	10
<a href="#">coap_server_path</a>		
URI path structure	. . . . .	11
<a href="#">coap_server_path_list_t</a>		
URI path list structure	. . . . .	12
<a href="#">coap_server_trans</a>		
Transaction structure	. . . . .	13



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">lib/include/coap_client.h</a>	
Include file for the FreeCoAP client library . . . . .	17
<a href="#">lib/include/coap_log.h</a>	
Include file for the FreeCoAP logging module . . . . .	20
<a href="#">lib/include/coap_msg.h</a>	
Include file for the FreeCoAP message parser/formatter library . . . . .	23
<a href="#">lib/include/coap_server.h</a>	
Include file for the FreeCoAP server library . . . . .	34
<a href="#">lib/src/coap_client.c</a>	
Source file for the FreeCoAP client library . . . . .	38
<a href="#">lib/src/coap_log.c</a>	
Source file for the FreeCoAP logging module . . . . .	40
<a href="#">lib/src/coap_msg.c</a>	
Source file for the FreeCoAP message parser/formatter library . . . . .	42
<a href="#">lib/src/coap_server.c</a>	
Source file for the FreeCoAP server library . . . . .	49



## Chapter 3

# Class Documentation

### 3.1 coap\_client\_t Struct Reference

Client structure.

```
#include <coap_client.h>
```

#### Public Attributes

- int [sd](#)
- int [timer\\_fd](#)
- struct timespec [timeout](#)
- unsigned [num\\_retrans](#)
- struct sockaddr\_in6 [server\\_sin](#)
- socklen\_t [server\\_sin\\_len](#)
- char [server\\_host](#) [COAP\_CLIENT\_HOST\_BUF\_LEN]
- char [server\\_port](#) [COAP\_CLIENT\_PORT\_BUF\_LEN]

#### 3.1.1 Detailed Description

Client structure.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 unsigned coap\_client\_t::num\_retrans

Current number of retransmissions

##### 3.1.2.2 int coap\_client\_t::sd

Socket descriptor

##### 3.1.2.3 char coap\_client\_t::server\_host[COAP\_CLIENT\_HOST\_BUF\_LEN]

String to hold the server host address

### 3.1.2.4 `char coap_client_t::server_port[COAP_CLIENT_PORT_BUF_LEN]`

String to hold the server port number

### 3.1.2.5 `struct sockaddr_in6 coap_client_t::server_sin`

IPv6 socket structure

### 3.1.2.6 `socklen_t coap_client_t::server_sin_len`

IPv6 socket structure length

### 3.1.2.7 `struct timespec coap_client_t::timeout`

Timeout value

### 3.1.2.8 `int coap_client_t::timer_fd`

Timer file descriptor

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

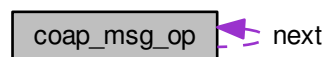
- [lib/include/coap\\_client.h](#)

## 3.2 `coap_msg_op` Struct Reference

Option structure.

```
#include <coap_msg.h>
```

Collaboration diagram for `coap_msg_op`:



### Public Attributes

- unsigned [num](#)
- unsigned [len](#)
- `char *` [val](#)
- `struct coap\_msg\_op *` [next](#)

### 3.2.1 Detailed Description

Option structure.

### 3.2.2 Member Data Documentation

#### 3.2.2.1 unsigned coap\_msg\_op::len

Option lenght

#### 3.2.2.2 struct coap\_msg\_op\* coap\_msg\_op::next

Pointer to the next option structure in the list

#### 3.2.2.3 unsigned coap\_msg\_op::num

Option number

#### 3.2.2.4 char\* coap\_msg\_op::val

Pointer to a buffer containing the option value

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

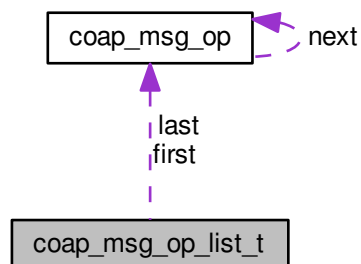
- lib/include/coap\_msg.h

## 3.3 coap\_msg\_op\_list\_t Struct Reference

Option linked-list structure.

```
#include <coap_msg.h>
```

Collaboration diagram for coap\_msg\_op\_list\_t:



### Public Attributes

- [coap\\_msg\\_op\\_t](#) \* first
- [coap\\_msg\\_op\\_t](#) \* last

### 3.3.1 Detailed Description

Option linked-list structure.

### 3.3.2 Member Data Documentation

#### 3.3.2.1 `coap_msg_op_t* coap_msg_op_list_t::first`

Pointer to the first option structure in the list

#### 3.3.2.2 `coap_msg_op_t* coap_msg_op_list_t::last`

Pointer to the last option structure in the list

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

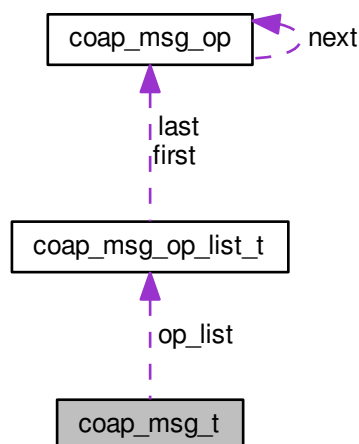
- `lib/include/coap_msg.h`

## 3.4 `coap_msg_t` Struct Reference

Message structure.

```
#include <coap_msg.h>
```

Collaboration diagram for `coap_msg_t`:



### Public Attributes

- unsigned `ver`
- `coap_msg_type_t` type
- unsigned `token_len`
- unsigned `code_class`
- unsigned `code_detail`
- unsigned `msg_id`
- char `token` [COAP\_MSG\_MAX\_TOKEN\_LEN]
- `coap_msg_op_list_t` `op_list`
- char \* `payload`
- size\_t `payload_len`



### 3.4.1 Detailed Description

Message structure.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 unsigned coap\_msg\_t::code\_class

Code class

#### 3.4.2.2 unsigned coap\_msg\_t::code\_detail

Code detail

#### 3.4.2.3 unsigned coap\_msg\_t::msg\_id

Message ID

#### 3.4.2.4 coap\_msg\_op\_list\_t coap\_msg\_t::op\_list

Option list

#### 3.4.2.5 char\* coap\_msg\_t::payload

Pointer to a buffer containing the payload

#### 3.4.2.6 size\_t coap\_msg\_t::payload\_len

Length of the payload

#### 3.4.2.7 char coap\_msg\_t::token[COAP\_MSG\_MAX\_TOKEN\_LEN]

Token value

#### 3.4.2.8 unsigned coap\_msg\_t::token\_len

Token length

#### 3.4.2.9 coap\_msg\_type\_t coap\_msg\_t::type

Message type

#### 3.4.2.10 unsigned coap\_msg\_t::ver

CoAP version

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

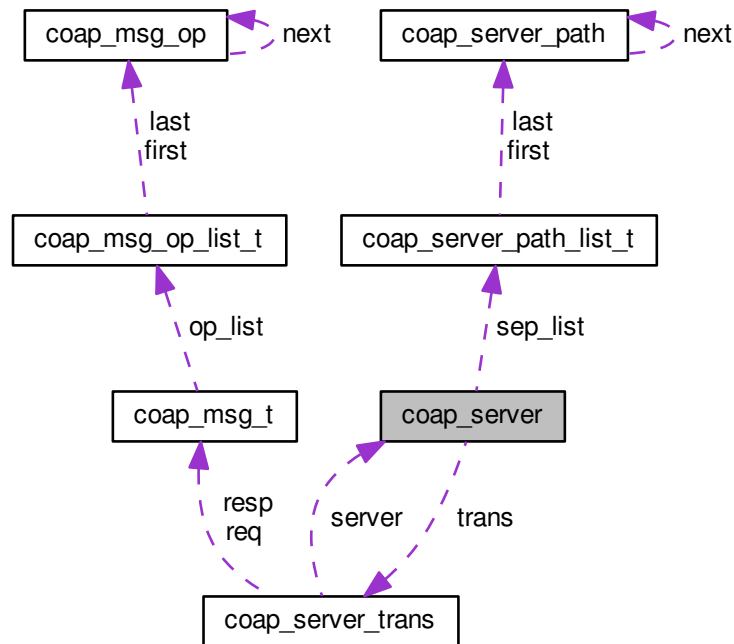
- lib/include/[coap\\_msg.h](#)

### 3.5 coap\_server Struct Reference

Server structure.

```
#include <coap_server.h>
```

Collaboration diagram for coap\_server:



#### Public Attributes

- `int sd`
- `unsigned msg_id`
- `coap_server_path_list_t sep_list`
- `coap_server_trans_t trans [COAP_SERVER_NUM_TRANS]`
- `int(* handle)(struct coap_server *, coap_msg_t *, coap_msg_t *)`

#### 3.5.1 Detailed Description

Server structure.

#### 3.5.2 Member Data Documentation

##### 3.5.2.1 `int(* coap_server::handle)(struct coap_server *, coap_msg_t *, coap_msg_t *)`

Call-back function to handle requests and generate responses

## 3.5.2.2 unsigned coap\_server::msg\_id

Last message ID value used in a response message

## 3.5.2.3 int coap\_server::sd

Socket descriptor

## 3.5.2.4 coap\_server\_path\_list\_t coap\_server::sep\_list

List of URI paths that require separate responses

## 3.5.2.5 coap\_server\_trans\_t coap\_server::trans[COAP\_SERVER\_NUM\_TRANS]

Array of transaction structures

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

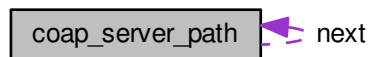
- [lib/include/coap\\_server.h](#)

## 3.6 coap\_server\_path Struct Reference

URI path structure.

```
#include <coap_server.h>
```

Collaboration diagram for coap\_server\_path:



### Public Attributes

- char \* [str](#)
- struct [coap\\_server\\_path](#) \* [next](#)

### 3.6.1 Detailed Description

URI path structure.

### 3.6.2 Member Data Documentation

## 3.6.2.1 struct coap\_server\_path\* coap\_server\_path::next

Pointer to the next URI path structure in the list

### 3.6.2.2 char\* coap\_server\_path::str

String containing a path

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

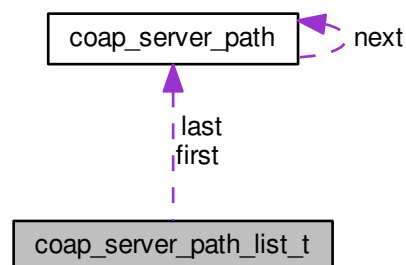
- lib/include/coap\_server.h

## 3.7 coap\_server\_path\_list\_t Struct Reference

URI path list structure.

```
#include <coap_server.h>
```

Collaboration diagram for coap\_server\_path\_list\_t:



### Public Attributes

- [coap\\_server\\_path\\_t \\* first](#)
- [coap\\_server\\_path\\_t \\* last](#)

### 3.7.1 Detailed Description

URI path list structure.

### 3.7.2 Member Data Documentation

#### 3.7.2.1 coap\_server\_path\_t\* coap\_server\_path\_list\_t::first

Pointer to the first URI path structure in the list

#### 3.7.2.2 coap\_server\_path\_t\* coap\_server\_path\_list\_t::last

Pointer to the last URI path structure in the list

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

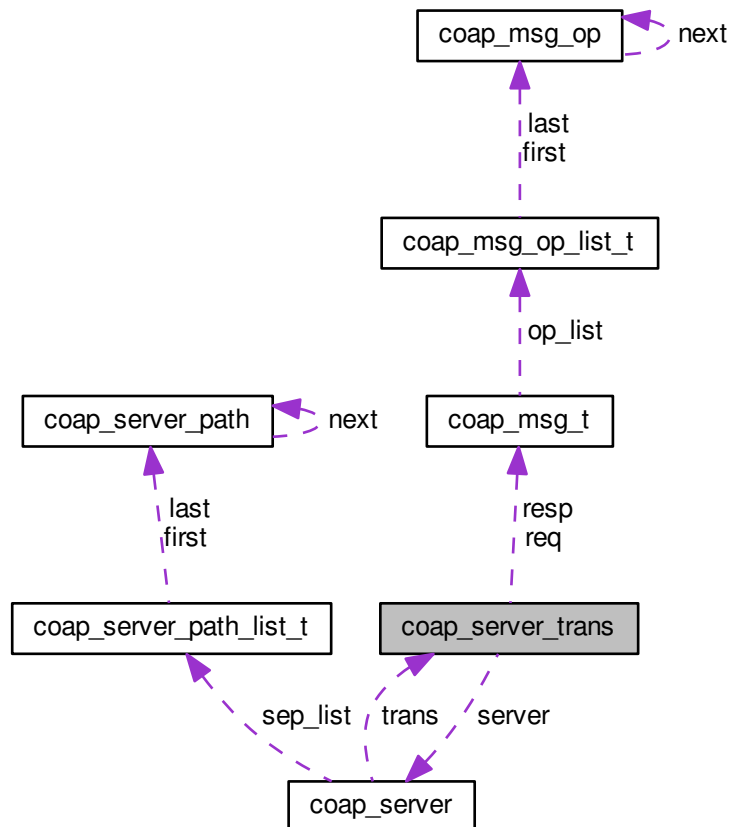
- lib/include/coap\_server.h

## 3.8 coap\_server\_trans Struct Reference

Transaction structure.

```
#include <coap_server.h>
```

Collaboration diagram for coap\_server\_trans:



### Public Attributes

- int `active`
- time\_t `last_use`
- int `timer_fd`
- struct timespec `timeout`
- unsigned `num_retrans`
- struct sockaddr\_in6 `client_sin`
- socklen\_t `client_sin_len`
- char `client_addr` [COAP\_SERVER\_ADDR\_BUF\_LEN]
- coap\_msg\_t `req`
- coap\_msg\_t `resp`
- struct coap\_server \* `server`

### 3.8.1 Detailed Description

Transaction structure.

### 3.8.2 Member Data Documentation

#### 3.8.2.1 `int coap_server_trans::active`

Flag to indicate if this transaction structure contains valid data

#### 3.8.2.2 `char coap_server_trans::client_addr[COAP_SERVER_ADDR_BUF_LEN]`

String to hold the client address

#### 3.8.2.3 `struct sockaddr_in6 coap_server_trans::client_sin`

IPv6 socket structure

#### 3.8.2.4 `socklen_t coap_server_trans::client_sin_len`

IPv6 socket structure length

#### 3.8.2.5 `time_t coap_server_trans::last_use`

The time that this transaction structure was last used

#### 3.8.2.6 `unsigned coap_server_trans::num_retrans`

Current number of retransmissions

#### 3.8.2.7 `coap_msg_t coap_server_trans::req`

Last request message received for this transaction

#### 3.8.2.8 `coap_msg_t coap_server_trans::resp`

Last response message sent for this transaction

#### 3.8.2.9 `struct coap_server* coap_server_trans::server`

Pointer to the containing server structure

#### 3.8.2.10 `struct timespec coap_server_trans::timeout`

Timeout value

#### 3.8.2.11 int coap\_server\_trans::timer\_fd

Timer file descriptor

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

- [lib/include/coap\\_server.h](#)





## Chapter 4

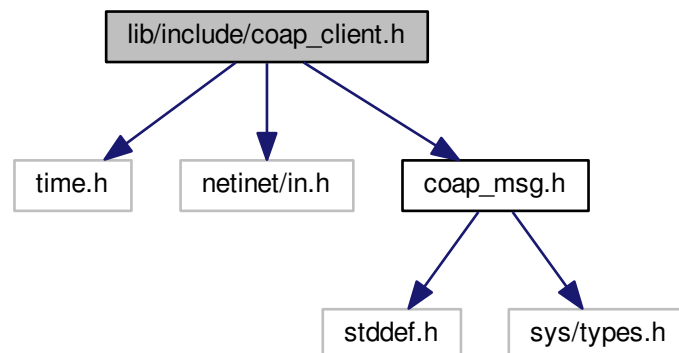
# File Documentation

### 4.1 lib/include/coap\_client.h File Reference

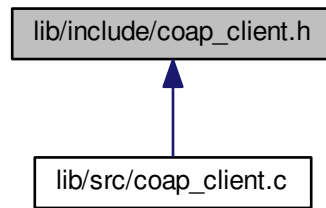
Include file for the FreeCoAP client library.

```
#include <time.h>
#include <netinet/in.h>
#include "coap_msg.h"
```

Include dependency graph for coap\_client.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct `coap_client_t`  
*Client structure.*

## Macros

- `#define COAP_CLIENT_HOST_BUF_LEN 128`
- `#define COAP_CLIENT_PORT_BUF_LEN 8`

## Functions

- int `coap_client_create` (`coap_client_t` \*client, const char \*host, const char \*port)  
*Initialise a client structure.*
- void `coap_client_destroy` (`coap_client_t` \*client)  
*Deinitialise a client structure.*
- int `coap_client_exchange` (`coap_client_t` \*client, `coap_msg_t` \*req, `coap_msg_t` \*resp)  
*Send a request to the server and receive the response.*

### 4.1.1 Detailed Description

Include file for the FreeCoAP client library.

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 `#define COAP_CLIENT_HOST_BUF_LEN 128`

Buffer length for host addresses

#### 4.1.2.2 `#define COAP_CLIENT_PORT_BUF_LEN 8`

Buffer length for port numbers

### 4.1.3 Function Documentation

4.1.3.1 int coap\_client\_create ( coap\_client\_t \* *client*, const char \* *host*, const char \* *port* )

Initialise a client structure.

**Parameters**

out	<i>client</i>	Pointer to a client structure
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server

**Returns**

Operation status

**Return values**

0	Success
<0	Error

**4.1.3.2 void coap\_client\_destroy ( coap\_client\_t \* client )**

Deinitialise a client structure.

**Parameters**

in, out	<i>client</i>	Pointer to a client structure
---------	---------------	-------------------------------

**4.1.3.3 int coap\_client\_exchange ( coap\_client\_t \* client, coap\_msg\_t \* req, coap\_msg\_t \* resp )**

Send a request to the server and receive the response.

**Parameters**

in, out	<i>client</i>	Pointer to a client structure
in	<i>req</i>	Pointer to the request message
out	<i>resp</i>	Pointer to the response message

This function sets the message ID and token fields of the request message overriding any values set by the calling function.

**Returns**

Operation status

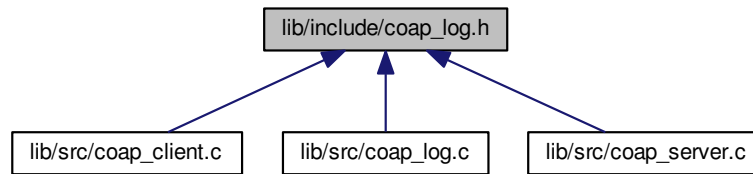
**Return values**

0	Success
<0	Error

**4.2 lib/include/coap\_log.h File Reference**

Include file for the FreeCoAP logging module.

This graph shows which files directly or indirectly include this file:



## Macros

- `#define COAP_LOG_DEF_LEVEL COAP_LOG_ERROR`

## Enumerations

- enum `coap_log_level_t` {  
`COAP_LOG_ERROR` = 0, `COAP_LOG_WARN` = 1, `COAP_LOG_NOTICE` = 2, `COAP_LOG_INFO` = 3,  
`COAP_LOG_DEBUG` = 4 }

*Log level.*

## Functions

- void `coap_log_set_level` (`coap_log_level_t` level)  
*Set the log level.*
- `coap_log_level_t` `coap_log_get_level` (void)  
*Get the log level.*
- void `coap_log_error` (const char \*msg,...)  
*Log an error message.*
- void `coap_log_warn` (const char \*msg,...)  
*Log a warning message.*
- void `coap_log_notice` (const char \*msg,...)  
*Log an notice message.*
- void `coap_log_info` (const char \*msg,...)  
*Log an info message.*
- void `coap_log_debug` (const char \*msg,...)  
*Log a debug message.*

### 4.2.1 Detailed Description

Include file for the FreeCoAP logging module.

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 `#define COAP_LOG_DEF_LEVEL COAP_LOG_ERROR`

Default log level

### 4.2.3 Enumeration Type Documentation

#### 4.2.3.1 enum coap\_log\_level\_t

Log level.

Enumerator

**COAP\_LOG\_ERROR** Highest severity level

**COAP\_LOG\_DEBUG** Lowest severity level

### 4.2.4 Function Documentation

#### 4.2.4.1 void coap\_log\_debug ( const char \* *msg*, ... )

Log a debug message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.2.4.2 void coap\_log\_error ( const char \* *msg*, ... )

Log an error message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.2.4.3 coap\_log\_level\_t coap\_log\_get\_level ( void )

Get the log level.

Returns

The current log level

#### 4.2.4.4 void coap\_log\_info ( const char \* *msg*, ... )

Log an info message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.2.4.5 void coap\_log\_notice ( const char \* *msg*, ... )

Log an notice message.

## Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

## 4.2.4.6 void coap\_log\_set\_level ( coap\_log\_level\_t level )

Set the log level.

Messages with a severity below this level will be filtered. Error messages cannot be filtered.

## Parameters

in	<i>level</i>	The new log level
----	--------------	-------------------

## 4.2.4.7 void coap\_log\_warn ( const char \* msg, ... )

Log a warning message.

## Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

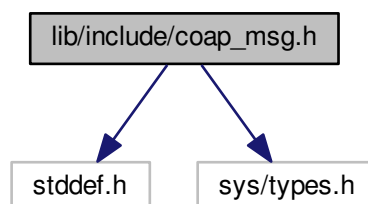
## 4.3 lib/include/coap\_msg.h File Reference

Include file for the FreeCoAP message parser/formatter library.

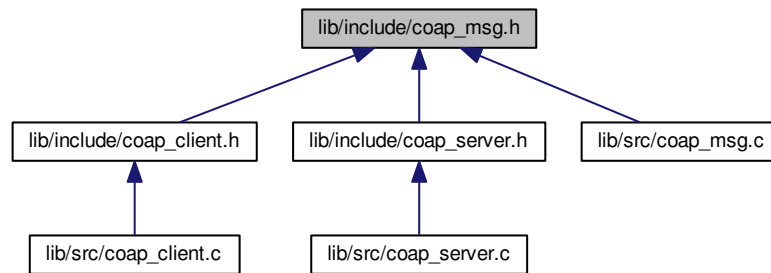
```
#include <stddef.h>
```

```
#include <sys/types.h>
```

Include dependency graph for coap\_msg.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct `coap_msg_op`  
*Option structure.*
- struct `coap_msg_op_list_t`  
*Option linked-list structure.*
- struct `coap_msg_t`  
*Message structure.*

## Macros

- `#define COAP_MSG_VER 0x01`
- `#define COAP_MSG_MAX_TOKEN_LEN 8`
- `#define COAP_MSG_MAX_CODE_CLASS 7`
- `#define COAP_MSG_MAX_CODE_DETAIL 31`
- `#define COAP_MSG_MAX_MSG_ID ((1 << 16) - 1)`
- `#define COAP_MSG_OP_URI_PATH_NUM 11`
- `#define COAP_MSG_OP_URI_PATH_MAX_LEN 256`
- `#define COAP_MSG_MAX_BUF_LEN 1152`
- `#define coap_msg_op_num_is_critical(num) ((num) & 1)`
- `#define coap_msg_op_num_is_unsafe(num) ((num) & 2)`
- `#define coap_msg_op_num_no_cache_key(num) ((num & 0x1e) == 0x1c)`
- `#define coap_msg_op_get_num(op) ((op)->num)`
- `#define coap_msg_op_set_num(op, num) ((op)->num = (num))`
- `#define coap_msg_op_get_len(op) ((op)->len)`
- `#define coap_msg_op_set_len(op, len) ((op)->len = (len))`
- `#define coap_msg_op_get_val(op) ((op)->val)`
- `#define coap_msg_op_set_val(op, val) ((op)->val = (val))`
- `#define coap_msg_op_get_next(op) ((op)->next)`
- `#define coap_msg_op_set_next(op, next_op) ((op)->next = (next_op))`
- `#define coap_msg_get_ver(msg) ((msg)->ver)`
- `#define coap_msg_get_type(msg) ((msg)->type)`
- `#define coap_msg_get_token_len(msg) ((msg)->token_len)`
- `#define coap_msg_get_code_class(msg) ((msg)->code_class)`
- `#define coap_msg_get_code_detail(msg) ((msg)->code_detail)`
- `#define coap_msg_get_msg_id(msg) ((msg)->msg_id)`
- `#define coap_msg_get_token(msg) ((msg)->token)`



- #define `coap_msg_get_first_op(msg)` `((msg)->op_list.first)`
- #define `coap_msg_get_payload(msg)` `((msg)->payload)`
- #define `coap_msg_get_payload_len(msg)` `((msg)->payload_len)`
- #define `coap_msg_is_empty(msg)` `((!(msg)->code_class == 0) && ((msg)->code_detail == 0))`

## Typedefs

- typedef struct `coap_msg_op` `coap_msg_op_t`  
*Option structure.*

## Enumerations

- enum `coap_msg_type_t` { `COAP_MSG_CON` = 0x0, `COAP_MSG_NON` = 0x1, `COAP_MSG_ACK` = 0x2, `COAP_MSG_RST` = 0x3 }  
*Message type enumeration.*
- enum `coap_msg_class_t` { `COAP_MSG_REQ` = 0, `COAP_MSG_SUCCESS` = 2, `COAP_MSG_CLIENT_ERR` = 4, `COAP_MSG_SERVER_ERR` = 5 }  
*Code class enumeration.*
- enum `coap_msg_method_t` { `COAP_MSG_GET` = 1, `COAP_MSG_POST` = 2, `COAP_MSG_PUT` = 3, `COAP_MSG_DELETE` = 4 }  
*Code detail enumeration.*
- enum `coap_msg_success_t` { `COAP_MSG_CREATED` = 1, `COAP_MSG_DELETED` = 2, `COAP_MSG_VALID` = 3, `COAP_MSG_CHANGED` = 4, `COAP_MSG_CONTENT` = 5 }  
*Success response code detail enumeration.*
- enum `coap_msg_client_err_t` { `COAP_MSG_BAD_REQ` = 0, `COAP_MSG_UNAUTHORIZED` = 1, `COAP_MSG_BAD_OPTION` = 2, `COAP_MSG_FORBIDDEN` = 3, `COAP_MSG_NOT_FOUND` = 4, `COAP_MSG_METHOD_NOT_ALLOWED` = 5, `COAP_MSG_NOT_ACCEPTABLE` = 6, `COAP_MSG_PRECOND_FAILED` = 12, `COAP_MSG_REQ_ENT_TOO_LARGE` = 13, `COAP_MSG_UNSUP_CONT_FMT` = 15 }  
*Client error response code detail enumeration.*
- enum `coap_msg_server_err_t` { `COAP_MSG_INT_SERVER_ERR` = 0, `COAP_MSG_NOT_IMPL` = 1, `COAP_MSG_BAD_GATEWAY` = 2, `COAP_MSG_SERV_UNAVAIL` = 3, `COAP_MSG_GATEWAY_TIMEOUT` = 4, `COAP_MSG_PROXY_NOT_SUP` = 5 }  
*Server error response code detail enumeration.*
- enum `coap_msg_op_num_t` { `COAP_MSG_IF_MATCH` = 1, `COAP_MSG_URI_HOST` = 3, `COAP_MSG_ETAG` = 4, `COAP_MSG_IF_NONE_MATCH` = 5, `COAP_MSG_URI_PORT` = 7, `COAP_MSG_LOCATION_PATH` = 8, `COAP_MSG_URI_PATH` = 11, `COAP_MSG_CONTENT_FORMAT` = 12, `COAP_MSG_MAX_AGE` = 14, `COAP_MSG_URI_QUERY` = 15, `COAP_MSG_ACCEPT` = 17, `COAP_MSG_LOCATION_QUERY` = 20, `COAP_MSG_PROXY_URI` = 35, `COAP_MSG_PROXY_SCHEME` = 39, `COAP_MSG_SIZE1` = 60 }  
*Option number enumeration.*

## Functions

- int `coap_msg_op_num_is_recognized` (unsigned num)  
*Check if option is recognized.*
- void `coap_msg_gen_rand_str` (char \*buf, size\_t len)

- Generate a random string of bytes.*

  - void `coap_msg_create` (`coap_msg_t *msg`)

*Initialise a message structure.*
- void `coap_msg_destroy` (`coap_msg_t *msg`)

*Deinitialise a message structure.*
- void `coap_msg_reset` (`coap_msg_t *msg`)

*Deinitialise and initialise a message structure.*
- unsigned `coap_msg_check_critical_ops` (`coap_msg_t *msg`)

*Check that all of the critical options in a message are recognized.*
- unsigned `coap_msg_check_unsafe_ops` (`coap_msg_t *msg`)

*Check that all of the unsafe options in a message are recognized.*
- int `coap_msg_parse_type_msg_id` (`char *buf`, `size_t len`, unsigned `*type`, unsigned `*msg_id`)

*Extract the type and message ID values from a message.*
- `ssize_t` `coap_msg_parse` (`coap_msg_t *msg`, `char *buf`, `size_t len`)

*Parse a message.*
- int `coap_msg_set_type` (`coap_msg_t *msg`, unsigned `type`)

*Set the type in a message.*
- int `coap_msg_set_code` (`coap_msg_t *msg`, unsigned `code_class`, unsigned `code_detail`)

*Set the code in a message.*
- int `coap_msg_set_msg_id` (`coap_msg_t *msg`, unsigned `msg_id`)

*Set the message ID in a message.*
- int `coap_msg_set_token` (`coap_msg_t *msg`, `char *buf`, `size_t len`)

*Set the token in a message.*
- int `coap_msg_add_op` (`coap_msg_t *msg`, unsigned `num`, unsigned `len`, const `char *val`)

*Add a token to a message structure.*
- int `coap_msg_set_payload` (`coap_msg_t *msg`, `char *buf`, `size_t len`)

*Set the payload in a message.*
- `ssize_t` `coap_msg_format` (`coap_msg_t *msg`, `char *buf`, `size_t len`)

*Format a message.*
- int `coap_msg_copy` (`coap_msg_t *dst`, `coap_msg_t *src`)

*Copy a message.*

### 4.3.1 Detailed Description

Include file for the FreeCoAP message parser/formatter library.

### 4.3.2 Macro Definition Documentation

#### 4.3.2.1 `#define coap_msg_get_code_class( msg ) ((msg)->code_class)`

Get the code class from a message

#### 4.3.2.2 `#define coap_msg_get_code_detail( msg ) ((msg)->code_detail)`

Get the code detail from a message

#### 4.3.2.3 `#define coap_msg_get_first_op( msg ) ((msg)->op_list.first)`

Get the first option from a message

4.3.2.4 `#define coap_msg_get_msg_id( msg ) ((msg)->msg_id)`

Get the message ID from message

4.3.2.5 `#define coap_msg_get_payload( msg ) ((msg)->payload)`

Get the payload from a message

4.3.2.6 `#define coap_msg_get_payload_len( msg ) ((msg)->payload_len)`

Get the payload length from a message

4.3.2.7 `#define coap_msg_get_token( msg ) ((msg)->token)`

Get the token from a message

4.3.2.8 `#define coap_msg_get_token_len( msg ) ((msg)->token_len)`

Get the token length from a message

4.3.2.9 `#define coap_msg_get_type( msg ) ((msg)->type)`

Get the type from a message

4.3.2.10 `#define coap_msg_get_ver( msg ) ((msg)->ver)`

Get the version from a message

4.3.2.11 `#define coap_msg_is_empty( msg ) (((msg)->code_class == 0) && ((msg)->code_detail == 0))`

Indicate if a message is empty

4.3.2.12 `#define COAP_MSG_MAX_BUF_LEN 1152`

Maximum buffer length for header and payload

4.3.2.13 `#define COAP_MSG_MAX_CODE_CLASS 7`

Maximum code class

4.3.2.14 `#define COAP_MSG_MAX_CODE_DETAIL 31`

Maximum code detail

4.3.2.15 `#define COAP_MSG_MAX_MSG_ID ((1 << 16) - 1)`

Maximum message ID

#### 4.3.2.16 `#define COAP_MSG_MAX_TOKEN_LEN 8`

Maximum token length

#### 4.3.2.17 `#define coap_msg_op_get_len( op ) ((op)->len)`

Get the option length from an option

#### 4.3.2.18 `#define coap_msg_op_get_next( op ) ((op)->next)`

Get the next pointer from an option

#### 4.3.2.19 `#define coap_msg_op_get_num( op ) ((op)->num)`

Get the option number from an option

#### 4.3.2.20 `#define coap_msg_op_get_val( op ) ((op)->val)`

Get the option value from an option

#### 4.3.2.21 `#define coap_msg_op_num_is_critical( num ) ((num) & 1)`

Indicate if an option is critical

#### 4.3.2.22 `#define coap_msg_op_num_is_unsafe( num ) ((num) & 2)`

Indicate if an option is unsafe to forward

#### 4.3.2.23 `#define coap_msg_op_num_no_cache_key( num ) ((num & 0x1e) == 0x1c)`

Indicate if an option is not part of the cache key

#### 4.3.2.24 `#define coap_msg_op_set_len( op, len ) ((op)->len = (len))`

Set the option length in an option

#### 4.3.2.25 `#define coap_msg_op_set_next( op, next_op ) ((op)->next = (next_op))`

Set the next pointer in an option

#### 4.3.2.26 `#define coap_msg_op_set_num( op, num ) ((op)->num = (num))`

Set the option number in an option

#### 4.3.2.27 `#define coap_msg_op_set_val( op, val ) ((op)->val = (val))`

Set the option value in an option

4.3.2.28 `#define COAP_MSG_OP_URI_PATH_MAX_LEN 256`

Maximum buffer length for a reconstructed URI path

4.3.2.29 `#define COAP_MSG_OP_URI_PATH_NUM 11`

Uri-path option number

4.3.2.30 `#define COAP_MSG_VER 0x01`

CoAP version

### 4.3.3 Function Documentation

4.3.3.1 `int coap_msg_add_op ( coap_msg_t * msg, unsigned num, unsigned len, const char * val )`

Add a token to a message structure.

#### Parameters

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
<i>in</i>	<i>num</i>	Option number
<i>in</i>	<i>len</i>	Option length
<i>in</i>	<i>val</i>	Pointer to a buffer containing the option value

#### Returns

Operation status

#### Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

4.3.3.2 `unsigned coap_msg_check_critical_ops ( coap_msg_t * msg )`

Check that all of the critical options in a message are recognized.

#### Parameters

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

#### Returns

Operation status or bad option number

#### Return values

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

4.3.3.3 `unsigned coap_msg_check_unsafe_ops ( coap_msg_t * msg )`

Check that all of the unsafe options in a message are recognized.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

**Returns**

Operation status or bad option number

**Return values**

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

**4.3.3.4 int coap\_msg\_copy ( coap\_msg\_t \* dst, coap\_msg\_t \* src )**

Copy a message.

**Parameters**

<i>in, out</i>	<i>dst</i>	Pointer to the destination message structure
<i>in</i>	<i>src</i>	Pointer to the source message structure

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.3.3.5 void coap\_msg\_create ( coap\_msg\_t \* msg )**

Initialise a message structure.

**Parameters**

<i>out</i>	<i>msg</i>	Pointer to a message structure
------------	------------	--------------------------------

**4.3.3.6 void coap\_msg\_destroy ( coap\_msg\_t \* msg )**

Deinitialise a message structure.

**Parameters**

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
----------------	------------	--------------------------------

**4.3.3.7 ssize\_t coap\_msg\_format ( coap\_msg\_t \* msg, char \* buf, size\_t len )**

Format a message.

**Parameters**


---

in	<i>msg</i>	Pointer to a message structure
out	<i>buf</i>	Pointer to a buffer to contain the formatted message
in	<i>len</i>	Length of the buffer

**Returns**

Length of the formatted message or error code

**Return values**

>0	Length of the formatted message
<0	Error

**4.3.3.8 void coap\_msg\_gen\_rand\_str ( char \* *buf*, size\_t *len* )**

Generate a random string of bytes.

**Parameters**

out	<i>buf</i>	Pointer to the buffer to store the random string
in	<i>len</i>	Length of the buffer

**4.3.3.9 int coap\_msg\_op\_num\_is\_recognized ( unsigned *num* )**

Check if option is recognized.

**Parameters**

in	<i>num</i>	Option number
----	------------	---------------

**Returns**

Operation status

**Return values**

1	Option is recognized
0	Option is not recognized

**4.3.3.10 ssize\_t coap\_msg\_parse ( coap\_msg\_t \* *msg*, char \* *buf*, size\_t *len* )**

Parse a message.

**Parameters**

in, out	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer

**Returns**

Operation status

## Return values

0	Success
<0	Error

## 4.3.3.11 int coap\_msg\_parse\_type\_msg\_id ( char \* buf, size\_t len, unsigned \* type, unsigned \* msg\_id )

Extract the type and message ID values from a message.

If a message contains a format error, this function will attempt to extract the type and message ID so that a reset message can be returned to the sender.

## Parameters

in	buf	Pointer to a buffer containing the message
in	len	Length of the buffer
out	type	Pointer to field to store the type value
out	msg_id	Pointer to a field to store the message ID value

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.3.3.12 void coap\_msg\_reset ( coap\_msg\_t \* msg )

Deinitialise and initialise a message structure.

## Parameters

in, out	msg	Pointer to a message structure
---------	-----	--------------------------------

## 4.3.3.13 int coap\_msg\_set\_code ( coap\_msg\_t \* msg, unsigned code\_class, unsigned code\_detail )

Set the code in a message.

## Parameters

out	msg	Pointer to a message structure
in	code_class	Code class
in	code_detail	Code detail

## Returns

Operation status

## Return values

0	Success
---	---------



<0	Error
----	-------

#### 4.3.3.14 int coap\_msg\_set\_msg\_id ( coap\_msg\_t \* msg, unsigned msg\_id )

Set the message ID in a message.

##### Parameters

out	msg	Pointer to a message structure
in	msg_id	Message ID

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.3.3.15 int coap\_msg\_set\_payload ( coap\_msg\_t \* msg, char \* buf, size\_t len )

Set the payload in a message.

Free the buffer in the message structure containing the current payload if there is one, allocate a buffer to contain the new payload and copy the buffer argument into the new payload buffer.

##### Parameters

in, out	msg	Pointer to a message structure
in	buf	Pointer to a buffer containing the payload
in	len	Length of the buffer

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.3.3.16 int coap\_msg\_set\_token ( coap\_msg\_t \* msg, char \* buf, size\_t len )

Set the token in a message.

##### Parameters

out	msg	Pointer to a message structure
in	buf	Pointer to a buffer containing the token
in	len	Length of the buffer

##### Returns

Operation status

## Return values

	0	Success
	<0	Error

## 4.3.3.17 int coap\_msg\_set\_type ( coap\_msg\_t \* msg, unsigned type )

Set the type in a message.

## Parameters

out	msg	Pointer to a message structure
in	type	Message type

## Returns

Operation status

## Return values

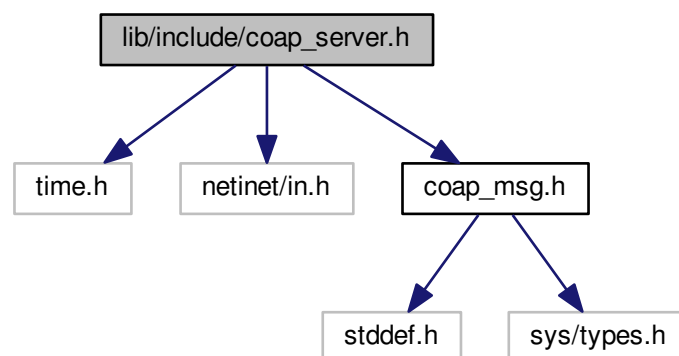
	0	Success
	<0	Error

## 4.4 lib/include/coap\_server.h File Reference

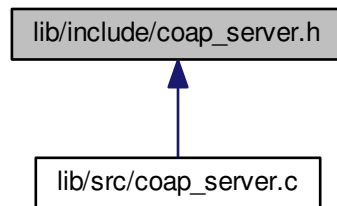
Include file for the FreeCoAP server library.

```
#include <time.h>
#include <netinet/in.h>
#include "coap_msg.h"
```

Include dependency graph for coap\_server.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [coap\\_server\\_path](#)  
*URI path structure.*
- struct [coap\\_server\\_path\\_list\\_t](#)  
*URI path list structure.*
- struct [coap\\_server\\_trans](#)  
*Transaction structure.*
- struct [coap\\_server](#)  
*Server structure.*

## Macros

- #define [COAP\\_SERVER\\_NUM\\_TRANS](#) 8
- #define [COAP\\_SERVER\\_ADDR\\_BUF\\_LEN](#) 128
- #define [COAP\\_SERVER\\_DIAG\\_PAYLOAD\\_LEN](#) 128

## Typedefs

- typedef struct [coap\\_server\\_path](#) [coap\\_server\\_path\\_t](#)  
*URI path structure.*
- typedef struct [coap\\_server\\_trans](#) [coap\\_server\\_trans\\_t](#)  
*Transaction structure.*
- typedef struct [coap\\_server](#) [coap\\_server\\_t](#)  
*Server structure.*

## Enumerations

- enum [coap\\_server\\_resp\\_t](#) { [COAP\\_SERVER\\_PIGGYBACKED](#) = 0, [COAP\\_SERVER\\_SEPARATE](#) = 1 }  
*Response type enumeration.*

## Functions

- int `coap_server_create` (`coap_server_t` \*server, int(\*handle)(`coap_server_t` \*, `coap_msg_t` \*, `coap_msg_t` \*), const char \*host, const char \*port)  
*Initialise a server structure.*
- void `coap_server_destroy` (`coap_server_t` \*server)  
*Deinitialise a server structure.*
- unsigned `coap_server_get_next_msg_id` (`coap_server_t` \*server)  
*Get a new message ID value.*
- int `coap_server_add_sep_resp_uri_path` (`coap_server_t` \*server, const char \*str)  
*Register a URI path that requires a separate response.*
- int `coap_server_run` (`coap_server_t` \*server)  
*Run the server.*

### 4.4.1 Detailed Description

Include file for the FreeCoAP server library.

### 4.4.2 Macro Definition Documentation

#### 4.4.2.1 #define COAP\_SERVER\_ADDR\_BUF\_LEN 128

Buffer length for host addresses

#### 4.4.2.2 #define COAP\_SERVER\_DIAG\_PAYLOAD\_LEN 128

Buffer length for diagnostic payloads

#### 4.4.2.3 #define COAP\_SERVER\_NUM\_TRANS 8

Maximum number of active transactions per server

### 4.4.3 Function Documentation

#### 4.4.3.1 int coap\_server\_add\_sep\_resp\_uri\_path ( coap\_server\_t \* server, const char \* str )

Register a URI path that requires a separate response.

Parameters

in, out	<i>server</i>	Pointer to a server structure
in	<i>str</i>	String representation of a URI path

Returns

Operation status

Return values

---

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.4.3.2** `int coap_server_create ( coap_server_t * server, int(*)(coap_server_t *, coap_msg_t *, coap_msg_t *)  
handle, const char * host, const char * port )`

Initialise a server structure.

#### Parameters

out	<i>server</i>	Pointer to a server structure
in	<i>handle</i>	Call-back function to handle client requests
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server

#### Returns

Operation status

#### Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.4.3.3** `void coap_server_destroy ( coap_server_t * server )`

Deinitialise a server structure.

#### Parameters

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

**4.4.3.4** `unsigned coap_server_get_next_msg_id ( coap_server_t * server )`

Get a new message ID value.

#### Parameters

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

#### Returns

message ID value

**4.4.3.5** `int coap_server_run ( coap_server_t * server )`

Run the server.

Listen for incoming requests. For each request received, call the handle call-back function in the server structure and send the response to the client.

#### Parameters

<code>in, out</code>	<code>server</code>	Pointer to a server structure
----------------------	---------------------	-------------------------------

**Returns**

Operation status

**Return values**

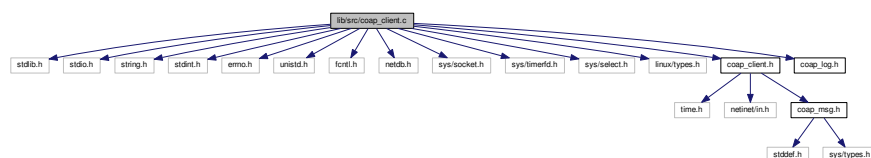
<code>0</code>	Success
<code>&lt;0</code>	Error

## 4.5 lib/src/coap\_client.c File Reference

Source file for the FreeCoAP client library.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/timerfd.h>
#include <sys/select.h>
#include <linux/types.h>
#include "coap_client.h"
#include "coap_log.h"
```

Include dependency graph for `coap_client.c`:

**Macros**

- `#define COAP_CLIENT_ACK_TIMEOUT_SEC 2`
- `#define COAP_CLIENT_MAX_RETRANSMIT 4`
- `#define COAP_CLIENT_RESP_TIMEOUT_SEC 30`

**Functions**

- `int coap_client_create (coap_client_t *client, const char *host, const char *port)`  
*Initialise a client structure.*
- `void coap_client_destroy (coap_client_t *client)`  
*Deinitialise a client structure.*
- `int coap_client_exchange (coap_client_t *client, coap_msg_t *req, coap_msg_t *resp)`  
*Send a request to the server and receive the response.*

### 4.5.1 Detailed Description

Source file for the FreeCoAP client library.

### 4.5.2 Macro Definition Documentation

#### 4.5.2.1 `#define COAP_CLIENT_ACK_TIMEOUT_SEC 2`

Minimum delay to wait before retransmitting a confirmable message

#### 4.5.2.2 `#define COAP_CLIENT_MAX_RETRANSMIT 4`

Maximum number of times a confirmable message can be retransmitted

#### 4.5.2.3 `#define COAP_CLIENT_RESP_TIMEOUT_SEC 30`

Maximum amount of time to wait for a response

### 4.5.3 Function Documentation

#### 4.5.3.1 `int coap_client_create ( coap_client_t * client, const char * host, const char * port )`

Initialise a client structure.

##### Parameters

out	<i>client</i>	Pointer to a client structure
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.5.3.2 `void coap_client_destroy ( coap_client_t * client )`

Deinitialise a client structure.

##### Parameters

in, out	<i>client</i>	Pointer to a client structure
---------	---------------	-------------------------------

#### 4.5.3.3 `int coap_client_exchange ( coap_client_t * client, coap_msg_t * req, coap_msg_t * resp )`

Send a request to the server and receive the response.

**Parameters**

in, out	<i>client</i>	Pointer to a client structure
in	<i>req</i>	Pointer to the request message
out	<i>resp</i>	Pointer to the response message

This function sets the message ID and token fields of the request message overriding any values set by the calling function.

**Returns**

Operation status

**Return values**

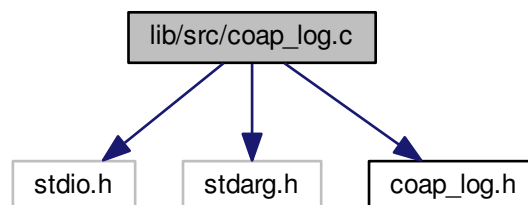
0	Success
<0	Error

## 4.6 lib/src/coap\_log.c File Reference

Source file for the FreeCoAP logging module.

```
#include <stdio.h>
#include <stdarg.h>
#include "coap_log.h"
```

Include dependency graph for coap\_log.c:

**Functions**

- void [coap\\_log\\_set\\_level](#) (coap\_log\_level\_t level)  
*Set the log level.*
- [coap\\_log\\_level\\_t coap\\_log\\_get\\_level](#) (void)  
*Get the log level.*
- void [coap\\_log\\_error](#) (const char \*msg,...)  
*Log an error message.*
- void [coap\\_log\\_warn](#) (const char \*msg,...)  
*Log a warning message.*
- void [coap\\_log\\_notice](#) (const char \*msg,...)  
*Log an notice message.*
- void [coap\\_log\\_info](#) (const char \*msg,...)  
*Log an info message.*



- void [coap\\_log\\_debug](#) (const char \*msg,...)  
*Log a debug message.*

### 4.6.1 Detailed Description

Source file for the FreeCoAP logging module.

### 4.6.2 Function Documentation

#### 4.6.2.1 void coap\_log\_debug ( const char \* msg, ... )

Log a debug message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.2 void coap\_log\_error ( const char \* msg, ... )

Log an error message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.3 coap\_log\_level\_t coap\_log\_get\_level ( void )

Get the log level.

Returns

The current log level

#### 4.6.2.4 void coap\_log\_info ( const char \* msg, ... )

Log an info message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.5 void coap\_log\_notice ( const char \* msg, ... )

Log an notice message.

Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.6 void coap\_log\_set\_level ( coap\_log\_level\_t level )

Set the log level.

Messages with a severity below this level will be filtered. Error messages cannot be filtered.

##### Parameters

in	<i>level</i>	The new log level
----	--------------	-------------------

#### 4.6.2.7 void coap\_log\_warn ( const char \* msg, ... )

Log a warning message.

##### Parameters

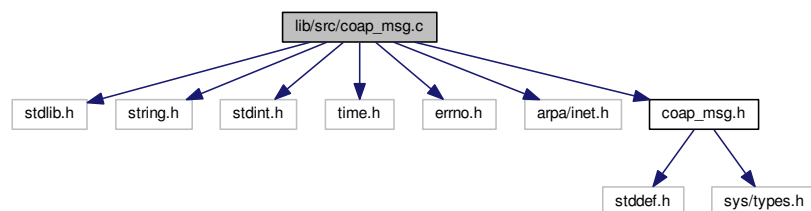
in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

## 4.7 lib/src/coap\_msg.c File Reference

Source file for the FreeCoAP message parser/formatter library.

```
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <time.h>
#include <errno.h>
#include <arpa/inet.h>
#include "coap_msg.h"
```

Include dependency graph for coap\_msg.c:



## Macros

- #define `coap_msg_op_list_get_first(list)` ((list)->first)
- #define `coap_msg_op_list_get_last(list)` ((list)->last)
- #define `coap_msg_op_list_is_empty(list)` ((list)->first == NULL)

## Functions

- void [coap\\_msg\\_gen\\_rand\\_str](#) (char \*buf, size\_t len)  
*Generate a random string of bytes.*
- int [coap\\_msg\\_op\\_num\\_is\\_recognized](#) (unsigned num)  
*Check if option is recognized.*
- void [coap\\_msg\\_create](#) (coap\_msg\_t \*msg)  
*Initialise a message structure.*
- void [coap\\_msg\\_destroy](#) (coap\_msg\_t \*msg)  
*Deinitialise a message structure.*
- void [coap\\_msg\\_reset](#) (coap\_msg\_t \*msg)  
*Deinitialise and initialise a message structure.*
- unsigned [coap\\_msg\\_check\\_critical\\_ops](#) (coap\_msg\_t \*msg)  
*Check that all of the critical options in a message are recognized.*
- unsigned [coap\\_msg\\_check\\_unsafe\\_ops](#) (coap\_msg\_t \*msg)  
*Check that all of the unsafe options in a message are recognized.*
- int [coap\\_msg\\_parse\\_type\\_msg\\_id](#) (char \*buf, size\_t len, unsigned \*type, unsigned \*msg\_id)  
*Extract the type and message ID values from a message.*
- ssize\_t [coap\\_msg\\_parse](#) (coap\_msg\_t \*msg, char \*buf, size\_t len)  
*Parse a message.*
- int [coap\\_msg\\_set\\_type](#) (coap\_msg\_t \*msg, unsigned type)  
*Set the type in a message.*
- int [coap\\_msg\\_set\\_code](#) (coap\_msg\_t \*msg, unsigned code\_class, unsigned code\_detail)  
*Set the code in a message.*
- int [coap\\_msg\\_set\\_msg\\_id](#) (coap\_msg\_t \*msg, unsigned msg\_id)  
*Set the message ID in a message.*
- int [coap\\_msg\\_set\\_token](#) (coap\_msg\_t \*msg, char \*buf, size\_t len)  
*Set the token in a message.*
- int [coap\\_msg\\_add\\_op](#) (coap\_msg\_t \*msg, unsigned num, unsigned len, const char \*val)  
*Add a token to a message structure.*
- int [coap\\_msg\\_set\\_payload](#) (coap\_msg\_t \*msg, char \*buf, size\_t len)  
*Set the payload in a message.*
- ssize\_t [coap\\_msg\\_format](#) (coap\_msg\_t \*msg, char \*buf, size\_t len)  
*Format a message.*
- int [coap\\_msg\\_copy](#) (coap\_msg\_t \*dst, coap\_msg\_t \*src)  
*Copy a message.*

### 4.7.1 Detailed Description

Source file for the FreeCoAP message parser/formatter library.

### 4.7.2 Macro Definition Documentation

#### 4.7.2.1 #define coap\_msg\_op\_list\_get\_first( list ) ((list)->first)

Get the first option from an option linked-list

#### 4.7.2.2 #define coap\_msg\_op\_list\_get\_last( list ) ((list)->last)

Get the last option in an option linked-list

4.7.2.3 `#define coap_msg_op_list_is_empty( list ) ((list)->first == NULL)`

Indicate whether or not an option linked-list is empty

### 4.7.3 Function Documentation

4.7.3.1 `int coap_msg_add_op ( coap_msg_t * msg, unsigned num, unsigned len, const char * val )`

Add a token to a message structure.

#### Parameters

in, out	<i>msg</i>	Pointer to a message structure
in	<i>num</i>	Option number
in	<i>len</i>	Option length
in	<i>val</i>	Pointer to a buffer containing the option value

#### Returns

Operation status

#### Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

4.7.3.2 `unsigned coap_msg_check_critical_ops ( coap_msg_t * msg )`

Check that all of the critical options in a message are recognized.

#### Parameters

in	<i>msg</i>	Pointer to message structure
----	------------	------------------------------

#### Returns

Operation status or bad option number

#### Return values

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

4.7.3.3 `unsigned coap_msg_check_unsafe_ops ( coap_msg_t * msg )`

Check that all of the unsafe options in a message are recognized.

#### Parameters

in	<i>msg</i>	Pointer to message structure
----	------------	------------------------------

#### Returns

Operation status or bad option number

## Return values

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

4.7.3.4 int coap\_msg\_copy ( coap\_msg\_t \* *dst*, coap\_msg\_t \* *src* )

Copy a message.

## Parameters

<i>in, out</i>	<i>dst</i>	Pointer to the destination message structure
<i>in</i>	<i>src</i>	Pointer to the source message structure

## Returns

Operation status

## Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

4.7.3.5 void coap\_msg\_create ( coap\_msg\_t \* *msg* )

Initialise a message structure.

## Parameters

<i>out</i>	<i>msg</i>	Pointer to a message structure
------------	------------	--------------------------------

4.7.3.6 void coap\_msg\_destroy ( coap\_msg\_t \* *msg* )

Deinitialise a message structure.

## Parameters

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
----------------	------------	--------------------------------

4.7.3.7 ssize\_t coap\_msg\_format ( coap\_msg\_t \* *msg*, char \* *buf*, size\_t *len* )

Format a message.

## Parameters

<i>in</i>	<i>msg</i>	Pointer to a message structure
<i>out</i>	<i>buf</i>	Pointer to a buffer to contain the formatted message
<i>in</i>	<i>len</i>	Length of the buffer

## Returns

Length of the formatted message or error code

## Return values

$>0$	Length of the formatted message
$<0$	Error

## 4.7.3.8 void coap\_msg\_gen\_rand\_str ( char \* buf, size\_t len )

Generate a random string of bytes.

## Parameters

out	buf	Pointer to the buffer to store the random string
in	len	Length of the buffer

## 4.7.3.9 int coap\_msg\_op\_num\_is\_recognized ( unsigned num )

Check if option is recognized.

## Parameters

in	num	Option number
----	-----	---------------

## Returns

Operation status

## Return values

1	Option is recognized
0	Option is not recognized

## 4.7.3.10 ssize\_t coap\_msg\_parse ( coap\_msg\_t \* msg, char \* buf, size\_t len )

Parse a message.

## Parameters

in, out	msg	Pointer to a message structure
in	buf	Pointer to a buffer containing the message
in	len	Length of the buffer

## Returns

Operation status

## Return values

0	Success
$<0$	Error

## 4.7.3.11 int coap\_msg\_parse\_type\_msg\_id ( char \* buf, size\_t len, unsigned \* type, unsigned \* msg\_id )

Extract the type and message ID values from a message.

If a message contains a format error, this function will attempt to extract the type and message ID so that a reset message can be returned to the sender.

## Parameters

in	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer
out	<i>type</i>	Pointer to field to store the type value
out	<i>msg_id</i>	Pointer to a field to store the message ID value

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.7.3.12 void coap\_msg\_reset ( coap\_msg\_t \* msg )

Deinitialise and initialise a message structure.

## Parameters

in, out	<i>msg</i>	Pointer to a message structure
---------	------------	--------------------------------

## 4.7.3.13 int coap\_msg\_set\_code ( coap\_msg\_t \* msg, unsigned code\_class, unsigned code\_detail )

Set the code in a message.

## Parameters

out	<i>msg</i>	Pointer to a message structure
in	<i>code_class</i>	Code class
in	<i>code_detail</i>	Code detail

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.7.3.14 int coap\_msg\_set\_msg\_id ( coap\_msg\_t \* msg, unsigned msg\_id )

Set the message ID in a message.

## Parameters

out	<i>msg</i>	Pointer to a message structure
in	<i>msg_id</i>	Message ID

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.7.3.15 int coap\_msg\_set\_payload ( coap\_msg\_t \* msg, char \* buf, size\_t len )

Set the payload in a message.

Free the buffer in the message structure containing the current payload if there is one, allocate a buffer to contain the new payload and copy the buffer argument into the new payload buffer.

## Parameters

in, out	msg	Pointer to a message structure
in	buf	Pointer to a buffer containing the payload
in	len	Length of the buffer

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.7.3.16 int coap\_msg\_set\_token ( coap\_msg\_t \* msg, char \* buf, size\_t len )

Set the token in a message.

## Parameters

out	msg	Pointer to a message structure
in	buf	Pointer to a buffer containing the token
in	len	Length of the buffer

## Returns

Operation status

## Return values

0	Success
<0	Error

## 4.7.3.17 int coap\_msg\_set\_type ( coap\_msg\_t \* msg, unsigned type )

Set the type in a message.

## Parameters

out	msg	Pointer to a message structure
-----	-----	--------------------------------



in	type	Message type
----	------	--------------

**Returns**

Operation status

**Return values**

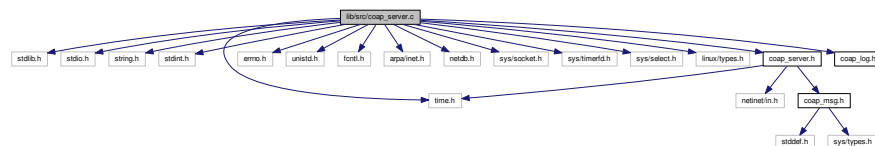
0	Success
<0	Error

## 4.8 lib/src/coap\_server.c File Reference

Source file for the FreeCoAP server library.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <time.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/timerfd.h>
#include <sys/select.h>
#include <linux/types.h>
#include "coap_server.h"
#include "coap_log.h"
```

Include dependency graph for coap\_server.c:

**Macros**

- `#define COAP_SERVER_ACK_TIMEOUT_SEC 2`
- `#define COAP_SERVER_MAX_RETRANSMIT 4`

**Functions**

- `int coap_server_create (coap_server_t *server, int(*handle)(coap_server_t *, coap_msg_t *, coap_msg_t *), const char *host, const char *port)`  
*Initialise a server structure.*
- `void coap_server_destroy (coap_server_t *server)`  
*Deinitialise a server structure.*
- `unsigned coap_server_get_next_msg_id (coap_server_t *server)`

*Get a new message ID value.*

- int `coap_server_add_sep_resp_uri_path` (`coap_server_t` \*server, const char \*str)

*Register a URI path that requires a separate response.*

- int `coap_server_run` (`coap_server_t` \*server)

*Run the server.*

#### 4.8.1 Detailed Description

Source file for the FreeCoAP server library.

#### 4.8.2 Macro Definition Documentation

##### 4.8.2.1 #define COAP\_SERVER\_ACK\_TIMEOUT\_SEC 2

Minimum delay to wait before retransmitting a confirmable message

##### 4.8.2.2 #define COAP\_SERVER\_MAX\_RETRANSMIT 4

Maximum number of times a confirmable message can be retransmitted

#### 4.8.3 Function Documentation

##### 4.8.3.1 int coap\_server\_add\_sep\_resp\_uri\_path ( coap\_server\_t \* server, const char \* str )

Register a URI path that requires a separate response.

Parameters

in, out	server	Pointer to a server structure
in	str	String representation of a URI path

Returns

Operation status

Return values

0	Success
<0	Error

##### 4.8.3.2 int coap\_server\_create ( coap\_server\_t \* server, int (\*)(coap\_server\_t \*, coap\_msg\_t \*, coap\_msg\_t \*) handle, const char \* host, const char \* port )

Initialise a server structure.

Parameters

out	server	Pointer to a server structure
in	handle	Call-back function to handle client requests
in	host	Pointer to a string containing the host address of the server

<i>in</i>	<i>port</i>	Port number of the server
-----------	-------------	---------------------------

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.8.3.3 void coap\_server\_destroy ( coap\_server\_t \* server )**

Deinitialise a server structure.

**Parameters**

<i>in, out</i>	<i>server</i>	Pointer to a server structure
----------------	---------------	-------------------------------

**4.8.3.4 unsigned coap\_server\_get\_next\_msg\_id ( coap\_server\_t \* server )**

Get a new message ID value.

**Parameters**

<i>in, out</i>	<i>server</i>	Pointer to a server structure
----------------	---------------	-------------------------------

**Returns**

message ID value

**4.8.3.5 int coap\_server\_run ( coap\_server\_t \* server )**

Run the server.

Listen for incoming requests. For each request received, call the handle call-back function in the server structure and send the response to the client.

**Parameters**

<i>in, out</i>	<i>server</i>	Pointer to a server structure
----------------	---------------	-------------------------------

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error



# Index

active

coap\_server\_trans, [14](#)

COAP\_CLIENT\_ACK\_TIMEOUT\_SEC

coap\_client.c, [39](#)

COAP\_CLIENT\_HOST\_BUF\_LEN

coap\_client.h, [18](#)

COAP\_CLIENT\_MAX\_RETRANSMIT

coap\_client.c, [39](#)

COAP\_CLIENT\_PORT\_BUF\_LEN

coap\_client.h, [18](#)

COAP\_CLIENT\_RESP\_TIMEOUT\_SEC

coap\_client.c, [39](#)

COAP\_LOG\_DEBUG

coap\_log.h, [22](#)

COAP\_LOG\_DEF\_LEVEL

coap\_log.h, [21](#)

COAP\_LOG\_ERROR

coap\_log.h, [22](#)

COAP\_MSG\_MAX\_BUF\_LEN

coap\_msg.h, [27](#)

COAP\_MSG\_MAX\_CODE\_CLASS

coap\_msg.h, [27](#)

COAP\_MSG\_MAX\_CODE\_DETAIL

coap\_msg.h, [27](#)

COAP\_MSG\_MAX\_MSG\_ID

coap\_msg.h, [27](#)

COAP\_MSG\_MAX\_TOKEN\_LEN

coap\_msg.h, [27](#)

COAP\_MSG\_OP\_URI\_PATH\_MAX\_LEN

coap\_msg.h, [28](#)

COAP\_MSG\_OP\_URI\_PATH\_NUM

coap\_msg.h, [29](#)

COAP\_MSG\_VER

coap\_msg.h, [29](#)

COAP\_SERVER\_ACK\_TIMEOUT\_SEC

coap\_server.c, [50](#)

COAP\_SERVER\_ADDR\_BUF\_LEN

coap\_server.h, [36](#)

COAP\_SERVER\_DIAG\_PAYLOAD\_LEN

coap\_server.h, [36](#)

COAP\_SERVER\_MAX\_RETRANSMIT

coap\_server.c, [50](#)

COAP\_SERVER\_NUM\_TRANS

coap\_server.h, [36](#)

client\_addr

coap\_server\_trans, [14](#)

client\_sin

coap\_server\_trans, [14](#)

client\_sin\_len

coap\_server\_trans, [14](#)

coap\_client.c

COAP\_CLIENT\_ACK\_TIMEOUT\_SEC, [39](#)

COAP\_CLIENT\_MAX\_RETRANSMIT, [39](#)

COAP\_CLIENT\_RESP\_TIMEOUT\_SEC, [39](#)

coap\_client\_create, [39](#)

coap\_client\_destroy, [39](#)

coap\_client\_exchange, [39](#)

coap\_client.h

COAP\_CLIENT\_HOST\_BUF\_LEN, [18](#)

COAP\_CLIENT\_PORT\_BUF\_LEN, [18](#)

coap\_client\_create, [19](#)

coap\_client\_destroy, [20](#)

coap\_client\_exchange, [20](#)

coap\_client\_create

coap\_client.c, [39](#)

coap\_client.h, [19](#)

coap\_client\_destroy

coap\_client.c, [39](#)

coap\_client.h, [20](#)

coap\_client\_exchange

coap\_client.c, [39](#)

coap\_client.h, [20](#)

coap\_client\_t, [5](#)

num\_retrans, [5](#)

sd, [5](#)

server\_host, [5](#)

server\_port, [5](#)

server\_sin, [6](#)

server\_sin\_len, [6](#)

timeout, [6](#)

timer\_fd, [6](#)

coap\_log.c

coap\_log\_debug, [41](#)

coap\_log\_error, [41](#)

coap\_log\_get\_level, [41](#)

coap\_log\_info, [41](#)

coap\_log\_notice, [41](#)

coap\_log\_set\_level, [42](#)

coap\_log\_warn, [42](#)

coap\_log.h

COAP\_LOG\_DEBUG, [22](#)

COAP\_LOG\_DEF\_LEVEL, [21](#)

COAP\_LOG\_ERROR, [22](#)

coap\_log\_debug, [22](#)

coap\_log\_error, [22](#)

coap\_log\_get\_level, [22](#)

coap\_log\_info, [22](#)

coap\_log\_level\_t, [22](#)

- coap\_log\_notice, 22
- coap\_log\_set\_level, 23
- coap\_log\_warn, 23
- coap\_log\_debug
  - coap\_log.c, 41
  - coap\_log.h, 22
- coap\_log\_error
  - coap\_log.c, 41
  - coap\_log.h, 22
- coap\_log\_get\_level
  - coap\_log.c, 41
  - coap\_log.h, 22
- coap\_log\_info
  - coap\_log.c, 41
  - coap\_log.h, 22
- coap\_log\_level\_t
  - coap\_log.h, 22
- coap\_log\_notice
  - coap\_log.c, 41
  - coap\_log.h, 22
- coap\_log\_set\_level
  - coap\_log.c, 42
  - coap\_log.h, 23
- coap\_log\_warn
  - coap\_log.c, 42
  - coap\_log.h, 23
- coap\_msg.c
  - coap\_msg\_add\_op, 44
  - coap\_msg\_check\_critical\_ops, 44
  - coap\_msg\_check\_unsafe\_ops, 44
  - coap\_msg\_copy, 45
  - coap\_msg\_create, 45
  - coap\_msg\_destroy, 45
  - coap\_msg\_format, 45
  - coap\_msg\_gen\_rand\_str, 46
  - coap\_msg\_op\_list\_get\_first, 43
  - coap\_msg\_op\_list\_get\_last, 43
  - coap\_msg\_op\_list\_is\_empty, 43
  - coap\_msg\_op\_num\_is\_recognized, 46
  - coap\_msg\_parse, 46
  - coap\_msg\_parse\_type\_msg\_id, 46
  - coap\_msg\_reset, 47
  - coap\_msg\_set\_code, 47
  - coap\_msg\_set\_msg\_id, 47
  - coap\_msg\_set\_payload, 48
  - coap\_msg\_set\_token, 48
  - coap\_msg\_set\_type, 48
- coap\_msg.h
  - COAP\_MSG\_MAX\_BUF\_LEN, 27
  - COAP\_MSG\_MAX\_CODE\_CLASS, 27
  - COAP\_MSG\_MAX\_CODE\_DETAIL, 27
  - COAP\_MSG\_MAX\_MSG\_ID, 27
  - COAP\_MSG\_MAX\_TOKEN\_LEN, 27
  - COAP\_MSG\_OP\_URI\_PATH\_MAX\_LEN, 28
  - COAP\_MSG\_OP\_URI\_PATH\_NUM, 29
  - COAP\_MSG\_VER, 29
  - coap\_msg\_add\_op, 29
  - coap\_msg\_check\_critical\_ops, 29
  - coap\_msg\_check\_unsafe\_ops, 29
  - coap\_msg\_copy, 30
  - coap\_msg\_create, 30
  - coap\_msg\_destroy, 30
  - coap\_msg\_format, 30
  - coap\_msg\_gen\_rand\_str, 31
  - coap\_msg\_get\_code\_class, 26
  - coap\_msg\_get\_code\_detail, 26
  - coap\_msg\_get\_first\_op, 26
  - coap\_msg\_get\_msg\_id, 26
  - coap\_msg\_get\_payload, 27
  - coap\_msg\_get\_payload\_len, 27
  - coap\_msg\_get\_token, 27
  - coap\_msg\_get\_token\_len, 27
  - coap\_msg\_get\_type, 27
  - coap\_msg\_get\_ver, 27
  - coap\_msg\_is\_empty, 27
  - coap\_msg\_op\_get\_len, 28
  - coap\_msg\_op\_get\_next, 28
  - coap\_msg\_op\_get\_num, 28
  - coap\_msg\_op\_get\_val, 28
  - coap\_msg\_op\_num\_is\_critical, 28
  - coap\_msg\_op\_num\_is\_recognized, 31
  - coap\_msg\_op\_num\_is\_unsafe, 28
  - coap\_msg\_op\_num\_no\_cache\_key, 28
  - coap\_msg\_op\_set\_len, 28
  - coap\_msg\_op\_set\_next, 28
  - coap\_msg\_op\_set\_num, 28
  - coap\_msg\_op\_set\_val, 28
  - coap\_msg\_parse, 31
  - coap\_msg\_parse\_type\_msg\_id, 32
  - coap\_msg\_reset, 32
  - coap\_msg\_set\_code, 32
  - coap\_msg\_set\_msg\_id, 33
  - coap\_msg\_set\_payload, 33
  - coap\_msg\_set\_token, 33
  - coap\_msg\_set\_type, 34
- coap\_msg\_add\_op
  - coap\_msg.c, 44
  - coap\_msg.h, 29
- coap\_msg\_check\_critical\_ops
  - coap\_msg.c, 44
  - coap\_msg.h, 29
- coap\_msg\_check\_unsafe\_ops
  - coap\_msg.c, 44
  - coap\_msg.h, 29
- coap\_msg\_copy
  - coap\_msg.c, 45
  - coap\_msg.h, 30
- coap\_msg\_create
  - coap\_msg.c, 45
  - coap\_msg.h, 30
- coap\_msg\_destroy
  - coap\_msg.c, 45
  - coap\_msg.h, 30
- coap\_msg\_format
  - coap\_msg.c, 45
  - coap\_msg.h, 30

- coap\_msg\_gen\_rand\_str
  - coap\_msg.c, [46](#)
  - coap\_msg.h, [31](#)
- coap\_msg\_get\_code\_class
  - coap\_msg.h, [26](#)
- coap\_msg\_get\_code\_detail
  - coap\_msg.h, [26](#)
- coap\_msg\_get\_first\_op
  - coap\_msg.h, [26](#)
- coap\_msg\_get\_msg\_id
  - coap\_msg.h, [26](#)
- coap\_msg\_get\_payload
  - coap\_msg.h, [27](#)
- coap\_msg\_get\_payload\_len
  - coap\_msg.h, [27](#)
- coap\_msg\_get\_token
  - coap\_msg.h, [27](#)
- coap\_msg\_get\_token\_len
  - coap\_msg.h, [27](#)
- coap\_msg\_get\_type
  - coap\_msg.h, [27](#)
- coap\_msg\_get\_ver
  - coap\_msg.h, [27](#)
- coap\_msg\_is\_empty
  - coap\_msg.h, [27](#)
- coap\_msg\_op, [6](#)
  - len, [7](#)
  - next, [7](#)
  - num, [7](#)
  - val, [7](#)
- coap\_msg\_op\_get\_len
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_get\_next
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_get\_num
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_get\_val
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_list\_get\_first
  - coap\_msg.c, [43](#)
- coap\_msg\_op\_list\_get\_last
  - coap\_msg.c, [43](#)
- coap\_msg\_op\_list\_is\_empty
  - coap\_msg.c, [43](#)
- coap\_msg\_op\_list\_t, [7](#)
  - first, [8](#)
  - last, [8](#)
- coap\_msg\_op\_num\_is\_critical
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_num\_is\_recognized
  - coap\_msg.c, [46](#)
  - coap\_msg.h, [31](#)
- coap\_msg\_op\_num\_is\_unsafe
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_num\_no\_cache\_key
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_set\_len
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_set\_next
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_set\_num
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_set\_val
  - coap\_msg.h, [28](#)
- coap\_msg\_parse
  - coap\_msg.c, [46](#)
  - coap\_msg.h, [31](#)
- coap\_msg\_parse\_type\_msg\_id
  - coap\_msg.c, [46](#)
  - coap\_msg.h, [32](#)
- coap\_msg\_reset
  - coap\_msg.c, [47](#)
  - coap\_msg.h, [32](#)
- coap\_msg\_set\_code
  - coap\_msg.c, [47](#)
  - coap\_msg.h, [32](#)
- coap\_msg\_set\_msg\_id
  - coap\_msg.c, [47](#)
  - coap\_msg.h, [33](#)
- coap\_msg\_set\_payload
  - coap\_msg.c, [48](#)
  - coap\_msg.h, [33](#)
- coap\_msg\_set\_token
  - coap\_msg.c, [48](#)
  - coap\_msg.h, [33](#)
- coap\_msg\_set\_type
  - coap\_msg.c, [48](#)
  - coap\_msg.h, [34](#)
- coap\_msg\_t, [8](#)
  - code\_class, [9](#)
  - code\_detail, [9](#)
  - msg\_id, [9](#)
  - op\_list, [9](#)
  - payload, [9](#)
  - payload\_len, [9](#)
  - token, [9](#)
  - token\_len, [9](#)
  - type, [9](#)
  - ver, [9](#)
- coap\_server, [10](#)
  - handle, [10](#)
  - msg\_id, [10](#)
  - sd, [11](#)
  - sep\_list, [11](#)
  - trans, [11](#)
- coap\_server.c
  - COAP\_SERVER\_ACK\_TIMEOUT\_SEC, [50](#)
  - COAP\_SERVER\_MAX\_RETRANSMIT, [50](#)
  - coap\_server\_add\_sep\_resp\_uri\_path, [50](#)
  - coap\_server\_create, [50](#)
  - coap\_server\_destroy, [51](#)
  - coap\_server\_get\_next\_msg\_id, [51](#)
  - coap\_server\_run, [51](#)
- coap\_server.h
  - COAP\_SERVER\_ADDR\_BUF\_LEN, [36](#)
  - COAP\_SERVER\_DIAG\_PAYLOAD\_LEN, [36](#)

- COAP\_SERVER\_NUM\_TRANS, 36
- coap\_server\_add\_sep\_resp\_uri\_path, 36
- coap\_server\_create, 37
- coap\_server\_destroy, 37
- coap\_server\_get\_next\_msg\_id, 37
- coap\_server\_run, 37
- coap\_server\_add\_sep\_resp\_uri\_path
  - coap\_server.c, 50
  - coap\_server.h, 36
- coap\_server\_create
  - coap\_server.c, 50
  - coap\_server.h, 37
- coap\_server\_destroy
  - coap\_server.c, 51
  - coap\_server.h, 37
- coap\_server\_get\_next\_msg\_id
  - coap\_server.c, 51
  - coap\_server.h, 37
- coap\_server\_path, 11
  - next, 11
  - str, 11
- coap\_server\_path\_list\_t, 12
  - first, 12
  - last, 12
- coap\_server\_run
  - coap\_server.c, 51
  - coap\_server.h, 37
- coap\_server\_trans, 13
  - active, 14
  - client\_addr, 14
  - client\_sin, 14
  - client\_sin\_len, 14
  - last\_use, 14
  - num\_retrans, 14
  - req, 14
  - resp, 14
  - server, 14
  - timeout, 14
  - timer\_fd, 14
- code\_class
  - coap\_msg\_t, 9
- code\_detail
  - coap\_msg\_t, 9
- first
  - coap\_msg\_op\_list\_t, 8
  - coap\_server\_path\_list\_t, 12
- handle
  - coap\_server, 10
- last
  - coap\_msg\_op\_list\_t, 8
  - coap\_server\_path\_list\_t, 12
- last\_use
  - coap\_server\_trans, 14
- len
  - coap\_msg\_op, 7
- lib/include/coap\_client.h, 17
- lib/include/coap\_log.h, 20
- lib/include/coap\_msg.h, 23
- lib/include/coap\_server.h, 34
- lib/src/coap\_client.c, 38
- lib/src/coap\_log.c, 40
- lib/src/coap\_msg.c, 42
- lib/src/coap\_server.c, 49
- msg\_id
  - coap\_msg\_t, 9
  - coap\_server, 10
- next
  - coap\_msg\_op, 7
  - coap\_server\_path, 11
- num
  - coap\_msg\_op, 7
- num\_retrans
  - coap\_client\_t, 5
  - coap\_server\_trans, 14
- op\_list
  - coap\_msg\_t, 9
- payload
  - coap\_msg\_t, 9
- payload\_len
  - coap\_msg\_t, 9
- req
  - coap\_server\_trans, 14
- resp
  - coap\_server\_trans, 14
- sd
  - coap\_client\_t, 5
  - coap\_server, 11
- sep\_list
  - coap\_server, 11
- server
  - coap\_server\_trans, 14
- server\_host
  - coap\_client\_t, 5
- server\_port
  - coap\_client\_t, 5
- server\_sin
  - coap\_client\_t, 6
- server\_sin\_len
  - coap\_client\_t, 6
- str
  - coap\_server\_path, 11
- timeout
  - coap\_client\_t, 6
  - coap\_server\_trans, 14
- timer\_fd
  - coap\_client\_t, 6
  - coap\_server\_trans, 14
- token
  - coap\_msg\_t, 9



token\_len  
    coap\_msg\_t, [9](#)  
trans  
    coap\_server, [11](#)  
type  
    coap\_msg\_t, [9](#)  
  
val  
    coap\_msg\_op, [7](#)  
ver  
    coap\_msg\_t, [9](#)