

FreeCoAP

Generated by Doxygen 1.8.10

Wed Sep 2 2015 19:30:59

Contents

1	README	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	coap_client_t Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	7
4.1.2.1	num_retrans	7
4.1.2.2	sd	7
4.1.2.3	server_addr	7
4.1.2.4	server_sin	7
4.1.2.5	server_sin_len	8
4.1.2.6	timeout	8
4.1.2.7	timer_fd	8
4.2	coap_msg_op Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Member Data Documentation	8
4.2.2.1	len	8
4.2.2.2	next	8
4.2.2.3	num	8
4.2.2.4	val	9
4.3	coap_msg_op_list_t Struct Reference	9
4.3.1	Detailed Description	9
4.3.2	Member Data Documentation	9
4.3.2.1	first	9
4.3.2.2	last	9
4.4	coap_msg_t Struct Reference	9

4.4.1	Detailed Description	10
4.4.2	Member Data Documentation	10
4.4.2.1	code_class	10
4.4.2.2	code_detail	10
4.4.2.3	msg_id	10
4.4.2.4	op_list	10
4.4.2.5	payload	10
4.4.2.6	payload_len	10
4.4.2.7	token	10
4.4.2.8	token_len	10
4.4.2.9	type	10
4.4.2.10	ver	11
4.5	coap_server Struct Reference	11
4.5.1	Detailed Description	11
4.5.2	Member Data Documentation	11
4.5.2.1	handle	11
4.5.2.2	msg_id	11
4.5.2.3	sd	11
4.5.2.4	sep_list	11
4.5.2.5	trans	11
4.6	coap_server_path Struct Reference	12
4.6.1	Detailed Description	12
4.6.2	Member Data Documentation	12
4.6.2.1	next	12
4.6.2.2	str	12
4.7	coap_server_path_list_t Struct Reference	12
4.7.1	Detailed Description	12
4.7.2	Member Data Documentation	12
4.7.2.1	first	12
4.7.2.2	last	13
4.8	coap_server_trans Struct Reference	13
4.8.1	Detailed Description	13
4.8.2	Member Data Documentation	13
4.8.2.1	active	13
4.8.2.2	client_addr	13
4.8.2.3	client_sin	13
4.8.2.4	client_sin_len	13
4.8.2.5	last_use	14
4.8.2.6	num_retrans	14
4.8.2.7	req	14

4.8.2.8	resp	14
4.8.2.9	server	14
4.8.2.10	timeout	14
4.8.2.11	timer_fd	14
5	File Documentation	15
5.1	include/coap_client.h File Reference	15
5.1.1	Detailed Description	15
5.1.2	Macro Definition Documentation	15
5.1.2.1	COAP_CLIENT_ADDR_BUF_LEN	15
5.1.3	Function Documentation	16
5.1.3.1	coap_client_create(coap_client_t *client, const char *host, unsigned port)	16
5.1.3.2	coap_client_destroy(coap_client_t *client)	16
5.1.3.3	coap_client_exchange(coap_client_t *client, coap_msg_t *req, coap_msg_t *resp)	16
5.2	include/coap_log.h File Reference	16
5.2.1	Detailed Description	17
5.2.2	Macro Definition Documentation	17
5.2.2.1	COAP_LOG_DEF_LEVEL	17
5.2.3	Enumeration Type Documentation	17
5.2.3.1	coap_log_level_t	17
5.2.4	Function Documentation	17
5.2.4.1	coap_log_debug(const char *msg,...)	17
5.2.4.2	coap_log_error(const char *msg,...)	18
5.2.4.3	coap_log_get_level(void)	18
5.2.4.4	coap_log_info(const char *msg,...)	18
5.2.4.5	coap_log_notice(const char *msg,...)	18
5.2.4.6	coap_log_set_level(coap_log_level_t level)	18
5.2.4.7	coap_log_warn(const char *msg,...)	18
5.3	include/coap_msg.h File Reference	19
5.3.1	Detailed Description	21
5.3.2	Macro Definition Documentation	21
5.3.2.1	coap_msg_get_code_class	21
5.3.2.2	coap_msg_get_code_detail	21
5.3.2.3	coap_msg_get_first_op	21
5.3.2.4	coap_msg_get_msg_id	21
5.3.2.5	coap_msg_get_payload	21
5.3.2.6	coap_msg_get_payload_len	21
5.3.2.7	coap_msg_get_token	21
5.3.2.8	coap_msg_get_token_len	21
5.3.2.9	coap_msg_get_type	22

5.3.2.10	<code>coap_msg_get_ver</code>	22
5.3.2.11	<code>coap_msg_is_empty</code>	22
5.3.2.12	<code>COAP_MSG_MAX_BUF_LEN</code>	22
5.3.2.13	<code>COAP_MSG_MAX_CODE_CLASS</code>	22
5.3.2.14	<code>COAP_MSG_MAX_CODE_DETAIL</code>	22
5.3.2.15	<code>COAP_MSG_MAX_MSG_ID</code>	22
5.3.2.16	<code>COAP_MSG_MAX_TOKEN_LEN</code>	22
5.3.2.17	<code>coap_msg_op_get_len</code>	22
5.3.2.18	<code>coap_msg_op_get_next</code>	22
5.3.2.19	<code>coap_msg_op_get_num</code>	22
5.3.2.20	<code>coap_msg_op_get_val</code>	22
5.3.2.21	<code>coap_msg_op_num_is_critical</code>	23
5.3.2.22	<code>coap_msg_op_num_is_unsafe</code>	23
5.3.2.23	<code>coap_msg_op_num_no_cache_key</code>	23
5.3.2.24	<code>coap_msg_op_set_len</code>	23
5.3.2.25	<code>coap_msg_op_set_next</code>	23
5.3.2.26	<code>coap_msg_op_set_num</code>	23
5.3.2.27	<code>coap_msg_op_set_val</code>	23
5.3.2.28	<code>COAP_MSG_OP_URI_PATH_MAX_LEN</code>	23
5.3.2.29	<code>COAP_MSG_OP_URI_PATH_NUM</code>	23
5.3.2.30	<code>COAP_MSG_VER</code>	23
5.3.3	Function Documentation	23
5.3.3.1	<code>coap_msg_add_op(coap_msg_t *msg, unsigned num, unsigned len, const char *val)</code>	23
5.3.3.2	<code>coap_msg_copy(coap_msg_t *dst, coap_msg_t *src)</code>	24
5.3.3.3	<code>coap_msg_create(coap_msg_t *msg)</code>	24
5.3.3.4	<code>coap_msg_destroy(coap_msg_t *msg)</code>	24
5.3.3.5	<code>coap_msg_format(coap_msg_t *msg, char *buf, size_t len)</code>	24
5.3.3.6	<code>coap_msg_gen_rand_str(char *buf, size_t len)</code>	25
5.3.3.7	<code>coap_msg_parse(coap_msg_t *msg, char *buf, size_t len)</code>	25
5.3.3.8	<code>coap_msg_parse_type_msg_id(char *buf, size_t len, unsigned *type, unsigned *msg_id)</code>	25
5.3.3.9	<code>coap_msg_reset(coap_msg_t *msg)</code>	26
5.3.3.10	<code>coap_msg_set_code(coap_msg_t *msg, unsigned code_class, unsigned code←_detail)</code>	26
5.3.3.11	<code>coap_msg_set_msg_id(coap_msg_t *msg, unsigned msg_id)</code>	26
5.3.3.12	<code>coap_msg_set_payload(coap_msg_t *msg, char *buf, size_t len)</code>	27
5.3.3.13	<code>coap_msg_set_token(coap_msg_t *msg, char *buf, size_t len)</code>	27
5.3.3.14	<code>coap_msg_set_type(coap_msg_t *msg, unsigned type)</code>	27
5.4	<code>include/coap_server.h</code> File Reference	28
5.4.1	Detailed Description	29

5.4.2	Macro Definition Documentation	29
5.4.2.1	COAP_SERVER_ADDR_BUF_LEN	29
5.4.2.2	COAP_SERVER_NUM_TRANS	29
5.4.3	Function Documentation	29
5.4.3.1	coap_server_add_sep_resp_uri_path(coap_server_t *server, const char *str) . .	29
5.4.3.2	coap_server_create(coap_server_t *server, int(*handle)(coap_server_t *, coap_msg_t *, coap_msg_t *), const char *host, unsigned port)	29
5.4.3.3	coap_server_destroy(coap_server_t *server)	30
5.4.3.4	coap_server_get_next_msg_id(coap_server_t *server)	30
5.4.3.5	coap_server_run(coap_server_t *server)	30
5.5	src/coap_client.c File Reference	31
5.5.1	Detailed Description	31
5.5.2	Macro Definition Documentation	31
5.5.2.1	COAP_CLIENT_ACK_TIMEOUT_SEC	31
5.5.2.2	COAP_CLIENT_MAX_RETRANSMIT	31
5.5.2.3	COAP_CLIENT_RESP_TIMEOUT_SEC	32
5.5.3	Function Documentation	32
5.5.3.1	coap_client_create(coap_client_t *client, const char *host, unsigned port)	32
5.5.3.2	coap_client_destroy(coap_client_t *client)	32
5.5.3.3	coap_client_exchange(coap_client_t *client, coap_msg_t *req, coap_msg_t *resp) .	32
5.6	src/coap_msg.c File Reference	32
5.6.1	Detailed Description	33
5.6.2	Macro Definition Documentation	34
5.6.2.1	coap_msg_op_list_get_first	34
5.6.2.2	coap_msg_op_list_get_last	34
5.6.2.3	coap_msg_op_list_is_empty	34
5.6.3	Function Documentation	34
5.6.3.1	coap_msg_add_op(coap_msg_t *msg, unsigned num, unsigned len, const char *val)	34
5.6.3.2	coap_msg_copy(coap_msg_t *dst, coap_msg_t *src)	34
5.6.3.3	coap_msg_create(coap_msg_t *msg)	35
5.6.3.4	coap_msg_destroy(coap_msg_t *msg)	35
5.6.3.5	coap_msg_format(coap_msg_t *msg, char *buf, size_t len)	35
5.6.3.6	coap_msg_gen_rand_str(char *buf, size_t len)	35
5.6.3.7	coap_msg_parse(coap_msg_t *msg, char *buf, size_t len)	35
5.6.3.8	coap_msg_parse_type_msg_id(char *buf, size_t len, unsigned *type, unsigned *msg_id)	36
5.6.3.9	coap_msg_reset(coap_msg_t *msg)	36
5.6.3.10	coap_msg_set_code(coap_msg_t *msg, unsigned code_class, unsigned code↔ _detail)	36
5.6.3.11	coap_msg_set_msg_id(coap_msg_t *msg, unsigned msg_id)	37

5.6.3.12	<code>coap_msg_set_payload(coap_msg_t *msg, char *buf, size_t len)</code>	37
5.6.3.13	<code>coap_msg_set_token(coap_msg_t *msg, char *buf, size_t len)</code>	37
5.6.3.14	<code>coap_msg_set_type(coap_msg_t *msg, unsigned type)</code>	38
5.7	<code>src/coap_server.c</code> File Reference	38
5.7.1	Detailed Description	39
5.7.2	Macro Definition Documentation	39
5.7.2.1	<code>COAP_SERVER_ACK_TIMEOUT_SEC</code>	39
5.7.2.2	<code>COAP_SERVER_MAX_RETRANSMIT</code>	39
5.7.3	Function Documentation	39
5.7.3.1	<code>coap_server_add_sep_resp_uri_path(coap_server_t *server, const char *str)</code>	39
5.7.3.2	<code>coap_server_create(coap_server_t *server, int(*handle)(coap_server_t *, coap_msg_t *, coap_msg_t *), const char *host, unsigned port)</code>	39
5.7.3.3	<code>coap_server_destroy(coap_server_t *server)</code>	40
5.7.3.4	<code>coap_server_get_next_msg_id(coap_server_t *server)</code>	40
5.7.3.5	<code>coap_server_run(coap_server_t *server)</code>	40
Index		43

Chapter 1

README

=====

FreeCoAP

An implementation of the CoAP protocol for GNU/Linux consisting of:

- a CoAP message parser/formatter library
- a CoAP client library
- a CoAP server library
- unit test code for the message parser/formatter
- a client test application
- a server test application

Copyright (c) 2015 Keith Cullen

Released under a BSD style license.

Test Instructions

To build and test the message parser/formatter

```
$ cd test_msg
```

```
$ make
```

```
$ ./test_msg
```

To build and run the server test application

```
$ cd test_server
```

```
$ make
```

```
$ ./server
```

To build and run the client test application

```
$ cd test_client  
$ make  
$ ./client
```

To build and run the server test application with DTLS

```
$ cd test_server  
$ make dtls=y  
$ ./server
```

To build and run the client test application with DTLS

```
$ cd test_client  
$ make dtls=y  
$ ./client
```

Chapter 2

Class Index

2.1 Class List

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

coap_client_t		
Client structure	7
coap_msg_op		
Option structure	8
coap_msg_op_list_t		
Option linked-list structure	9
coap_msg_t		
Message structure	9
coap_server		
Server structure	11
coap_server_path		
URI path structure	12
coap_server_path_list_t		
URI path list structure	12
coap_server_trans		
Transaction structure	13

Chapter 3

File Index

3.1 File List

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

include/coap_client.h	
Include file for the FreeCoAP client library	15
include/coap_log.h	
Include file for the FreeCoAP logging module	16
include/coap_msg.h	
Include file for the FreeCoAP message parser/formatter library	19
include/coap_server.h	
Include file for the FreeCoAP server library	28
src/coap_client.c	
Source file for the FreeCoAP client library	31
src/coap_msg.c	
Source file for the FreeCoAP message parser/formatter library	32
src/coap_server.c	
Source file for the FreeCoAP server library	38

Chapter 4

Class Documentation

4.1 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_addr](#)[COAP_CLIENT_ADDR_BUF_LEN]

4.1.1 Detailed Description

Client structure.

4.1.2 Member Data Documentation

4.1.2.1 unsigned coap_client_t::num_retrans

Current number of retransmissions

4.1.2.2 int coap_client_t::sd

Socket descriptor

4.1.2.3 char coap_client_t::server_addr[COAP_CLIENT_ADDR_BUF_LEN]

String to hold the server address

4.1.2.4 struct sockaddr_in6 coap_client_t::server_sin

Ipv6 socket structure

4.1.2.5 socklen_t coap_client_t::server_sin_len

Ipv6 socket structure length

4.1.2.6 struct timespec coap_client_t::timeout

Timeout value

4.1.2.7 int coap_client_t::timer_fd

Timer file descriptor

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

- include/[coap_client.h](#)

4.2 coap_msg_op Struct Reference

Option structure.

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

Public Attributes

- unsigned [num](#)
- unsigned [len](#)
- char * [val](#)
- struct [coap_msg_op](#) * [next](#)

4.2.1 Detailed Description

Option structure.

4.2.2 Member Data Documentation

4.2.2.1 unsigned coap_msg_op::len

Option lenght

4.2.2.2 struct coap_msg_op* coap_msg_op::next

Pointer to the next option structure in the list

4.2.2.3 unsigned coap_msg_op::num

Option number

4.2.2.4 char* coap_msg_op::val

Pointer to a buffer containing the option value

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

- include/[coap_msg.h](#)

4.3 coap_msg_op_list_t Struct Reference

Option linked-list structure.

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

Public Attributes

- [coap_msg_op_t](#) * [first](#)
- [coap_msg_op_t](#) * [last](#)

4.3.1 Detailed Description

Option linked-list structure.

4.3.2 Member Data Documentation

4.3.2.1 coap_msg_op_t* coap_msg_op_list_t::first

Pointer to the first option structure in the list

4.3.2.2 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:

- include/[coap_msg.h](#)

4.4 coap_msg_t Struct Reference

Message structure.

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

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](#)

4.4.1 Detailed Description

Message structure.

4.4.2 Member Data Documentation

4.4.2.1 `unsigned coap_msg_t::code_class`

Code class

4.4.2.2 `unsigned coap_msg_t::code_detail`

Code detail

4.4.2.3 `unsigned coap_msg_t::msg_id`

Message ID

4.4.2.4 `coap_msg_op_list_t coap_msg_t::op_list`

Option list

4.4.2.5 `char* coap_msg_t::payload`

Pointer to a buffer containing the payload

4.4.2.6 `size_t coap_msg_t::payload_len`

Length of the payload

4.4.2.7 `char coap_msg_t::token[COAP_MSG_MAX_TOKEN_LEN]`

Token value

4.4.2.8 `unsigned coap_msg_t::token_len`

Token length

4.4.2.9 `coap_msg_type_t coap_msg_t::type`

Message type

4.4.2.10 unsigned coap_msg_t::ver

CoAP version

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

- [include/coap_msg.h](#)

4.5 coap_server Struct Reference

Server structure.

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

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](#) *)

4.5.1 Detailed Description

Server structure.

4.5.2 Member Data Documentation

4.5.2.1 int(* coap_server::handle) (struct coap_server *, coap_msg_t *, coap_msg_t *)

Call-back function to handle requests and generate responses

4.5.2.2 unsigned coap_server::msg_id

Last message ID value used in a response message

4.5.2.3 int coap_server::sd

Socket descriptor

4.5.2.4 coap_server_path_list_t coap_server::sep_list

List of URI paths that require separate responses

4.5.2.5 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:

- [include/coap_server.h](#)

4.6 coap_server_path Struct Reference

URI path structure.

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

Public Attributes

- char * [str](#)
- struct [coap_server_path](#) * [next](#)

4.6.1 Detailed Description

URI path structure.

4.6.2 Member Data Documentation

4.6.2.1 struct coap_server_path* coap_server_path::next

Pointer to the next URI path structure in the list

4.6.2.2 char* coap_server_path::str

String containing a path

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

- include/[coap_server.h](#)

4.7 coap_server_path_list_t Struct Reference

URI path list structure.

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

Public Attributes

- [coap_server_path_t](#) * [first](#)
- [coap_server_path_t](#) * [last](#)

4.7.1 Detailed Description

URI path list structure.

4.7.2 Member Data Documentation

4.7.2.1 coap_server_path_t* coap_server_path_list_t::first

Pointer to the first URI path structure in the list

4.7.2.2 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:

- include/[coap_server.h](#)

4.8 coap_server_trans Struct Reference

Transaction structure.

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

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](#)

4.8.1 Detailed Description

Transaction structure.

4.8.2 Member Data Documentation

4.8.2.1 int coap_server_trans::active

Flag to indicate if this transaction structure contains valid data

4.8.2.2 char coap_server_trans::client_addr[COAP_SERVER_ADDR_BUF_LEN]

String to hold the client address

4.8.2.3 struct sockaddr_in6 coap_server_trans::client_sin

Ipv6 socket structure

4.8.2.4 socklen_t coap_server_trans::client_sin_len

Ipv6 socket structure length

4.8.2.5 `time_t coap_server_trans::last_use`

The time that this transaction structure was last used

4.8.2.6 `unsigned coap_server_trans::num_retrans`

Current number of retransmissions

4.8.2.7 `coap_msg_t coap_server_trans::req`

Last request message received for this transaction

4.8.2.8 `coap_msg_t coap_server_trans::resp`

Last response message sent for this transaction

4.8.2.9 `struct coap_server* coap_server_trans::server`

Pointer to the containing server structure

4.8.2.10 `struct timespec coap_server_trans::timeout`

Timeout value

4.8.2.11 `int coap_server_trans::timer_fd`

Timer file descriptor

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

- [include/coap_server.h](#)

Chapter 5

File Documentation

5.1 include/coap_client.h File Reference

Include file for the FreeCoAP client library.

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

Classes

- struct [coap_client_t](#)
Client structure.

Macros

- #define [COAP_CLIENT_ADDR_BUF_LEN](#) 128

Functions

- int [coap_client_create](#) ([coap_client_t](#) *client, const char *host, unsigned 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.

5.1.1 Detailed Description

Include file for the FreeCoAP client library.

5.1.2 Macro Definition Documentation

5.1.2.1 #define COAP_CLIENT_ADDR_BUF_LEN 128

Buffer length for host addresses

5.1.3 Function Documentation

5.1.3.1 `int coap_client_create (coap_client_t * client, const char * host, unsigned 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

5.1.3.2 `void coap_client_destroy (coap_client_t * client)`

Deinitialise a client structure.

Parameters

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

5.1.3.3 `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

5.2 `include/coap_log.h` File Reference

Include file for the FreeCoAP logging module.

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.

5.2.1 Detailed Description

Include file for the FreeCoAP logging module.

Source file for the FreeCoAP logging module.

5.2.2 Macro Definition Documentation

5.2.2.1 #define COAP_LOG_DEF_LEVEL COAP_LOG_ERROR

Default log level

5.2.3 Enumeration Type Documentation

5.2.3.1 enum coap_log_level_t

Log level.

Enumerator

COAP_LOG_ERROR Highest severity level

COAP_LOG_DEBUG Lowest severity level

5.2.4 Function Documentation

5.2.4.1 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

5.2.4.2 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

5.2.4.3 coap_log_level_t coap_log_get_level (void)

Get the log level.

Returns

The current log level

5.2.4.4 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

5.2.4.5 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

5.2.4.6 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
----	--------------	-------------------

5.2.4.7 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

5.3 include/coap_msg.h File Reference

Include file for the FreeCoAP message parser/formatter library.

```
#include <stddef.h>
#include <linux/types.h>
```

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.

Functions

- 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.
- 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.

5.3.1 Detailed Description

Include file for the FreeCoAP message parser/formatter library.

5.3.2 Macro Definition Documentation

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

Get the code class from a message

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

Get the code detail from a message

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

Get the first option from a message

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

Get the message ID from message

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

Get the payload from a message

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

Get the payload length from a message

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

Get the token from a message

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

Get the token length from a message

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

Get the type from a message

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

Get the version from a message

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

Indicate if a message is empty

5.3.2.12 `#define COAP_MSG_MAX_BUF_LEN 1152`

Maximum buffer length for header and payload

5.3.2.13 `#define COAP_MSG_MAX_CODE_CLASS 7`

Maximum code class

5.3.2.14 `#define COAP_MSG_MAX_CODE_DETAIL 31`

Maximum code detail

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

Maximum message ID

5.3.2.16 `#define COAP_MSG_MAX_TOKEN_LEN 8`

Maximum token length

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

Get the option length from an option

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

Get the next pointer from an option

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

Get the option number from an option

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

Get the option value from an option

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

Indicate if an option is critical

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

Indicate if an option is unsafe to forward

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

Indicate if an option is not part of the cache key

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

Set the option length in an option

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

Set the next pointer in an option

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

Set the option number in an option

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

Set the option value in an option

5.3.2.28 `#define COAP_MSG_OP_URI_PATH_MAX_LEN 256`

Maximum buffer length for a reconstructed URI path

5.3.2.29 `#define COAP_MSG_OP_URI_PATH_NUM 11`

Uri-path option number

5.3.2.30 `#define COAP_MSG_VER 0x01`

CoAP version

5.3.3 Function Documentation

5.3.3.1 `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><0</i>	Error

5.3.3.2 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><0</i>	Error

5.3.3.3 void coap_msg_create (coap_msg_t * msg)

Initialise a message structure.

Parameters

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

5.3.3.4 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
----------------	------------	--------------------------------

5.3.3.5 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

5.3.3.6 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

5.3.3.7 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

5.3.3.8 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

5.3.3.9 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
---------	------------	--------------------------------

5.3.3.10 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

5.3.3.11 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

5.3.3.12 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

5.3.3.13 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

5.3.3.14 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

5.4 include/coap_server.h File Reference

Include file for the FreeCoAP server library.

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

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

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, unsigned 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.

5.4.1 Detailed Description

Include file for the FreeCoAP server library.

5.4.2 Macro Definition Documentation

5.4.2.1 #define COAP_SERVER_ADDR_BUF_LEN 128

Buffer length for host addresses

5.4.2.2 #define COAP_SERVER_NUM_TRANS 8

Maximum number of active transactions per server

5.4.3 Function Documentation

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

Returns

Operation status

Return values

0	Success
<0	Error

5.4.3.2 int coap_server_create (coap_server_t * server, int(*) (coap_server_t *, coap_msg_t *, coap_msg_t *) handle, const char * host, unsigned 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

0	Success
<0	Error

5.4.3.3 void coap_server_destroy (coap_server_t * server)

Deinitialise a server structure.

Parameters

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

5.4.3.4 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

5.4.3.5 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

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

Returns

Operation status

Return values

0	Success
<0	Error

5.5 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 <arpa/inet.h>
#include <sys/select.h>
#include <sys/timerfd.h>
#include <linux/types.h>
#include "coap_client.h"
#include "coap_log.h"
```

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, unsigned 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.

5.5.1 Detailed Description

Source file for the FreeCoAP client library.

5.5.2 Macro Definition Documentation

5.5.2.1 `#define COAP_CLIENT_ACK_TIMEOUT_SEC 2`

Minimum delay to wait before retransmitting a confirmable message

5.5.2.2 `#define COAP_CLIENT_MAX_RETRANSMIT 4`

Maximum number of times a confirmable message can be retransmitted

5.5.2.3 #define COAP_CLIENT_RESP_TIMEOUT_SEC 30

Maximum amount of time to wait for a response

5.5.3 Function Documentation

5.5.3.1 int coap_client_create (coap_client_t * *client*, const char * *host*, unsigned *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

5.5.3.2 void coap_client_destroy (coap_client_t * *client*)

Deinitialise a client structure.

Parameters

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

5.5.3.3 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

5.6 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"
```

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.
- 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.
- 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.

5.6.1 Detailed Description

Source file for the FreeCoAP message parser/formatter library.

5.6.2 Macro Definition Documentation

5.6.2.1 `#define coap_msg_op_list_get_first(list) ((list)->first)`

Get the first option from an option linked-list

5.6.2.2 `#define coap_msg_op_list_get_last(list) ((list)->last)`

Get the last option in an option linked-list

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

Indicate whether or not an option linked-list is empty

5.6.3 Function Documentation

5.6.3.1 `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><0</i>	Error

5.6.3.2 `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
----------	---------

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

5.6.3.3 void coap_msg_create (coap_msg_t * msg)

Initialise a message structure.

Parameters

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

5.6.3.4 void coap_msg_destroy (coap_msg_t * msg)

Deinitialise a message structure.

Parameters

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

5.6.3.5 ssize_t coap_msg_format (coap_msg_t * msg, char * buf, size_t len)

Format a message.

Parameters

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

Returns

Length of the formatted message or error code

Return values

>0	Length of the formatted message
<0	Error

5.6.3.6 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

5.6.3.7 ssize_t coap_msg_parse (coap_msg_t * msg, char * buf, size_t len)

Parse a message.

Parameters

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

Returns

Operation status

Return values

<i>0</i>	Success
<i><0</i>	Error

5.6.3.8 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

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

Returns

Operation status

Return values

<i>0</i>	Success
<i><0</i>	Error

5.6.3.9 void coap_msg_reset (coap_msg_t * *msg*)

Deinitialise and initialise a message structure.

Parameters

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

5.6.3.10 int coap_msg_set_code (coap_msg_t * *msg*, unsigned *code_class*, unsigned *code_detail*)

Set the code in a message.

Parameters

<i>out</i>	<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

5.6.3.11 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

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

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.13 int coap_msg_set_token (coap_msg_t * msg, char * buf, size_t len)

Set the token in a message.

Parameters

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

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.14 int coap_msg_set_type (coap_msg_t * msg, unsigned type)

Set the type in a message.

Parameters

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

Returns

Operation status

Return values

0	Success
<0	Error

5.7 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 <sys/select.h>
#include <sys/timerfd.h>
#include <linux/types.h>
#include "coap_server.h"
#include "coap_log.h"
```

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, unsigned 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.

5.7.1 Detailed Description

Source file for the FreeCoAP server library.

5.7.2 Macro Definition Documentation

5.7.2.1 #define COAP_SERVER_ACK_TIMEOUT_SEC 2

Minimum delay to wait before retransmitting a confirmable message

5.7.2.2 #define COAP_SERVER_MAX_RETRANSMIT 4

Maximum number of times a confirmable message can be retransmitted

5.7.3 Function Documentation

5.7.3.1 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

0	Success
<0	Error

5.7.3.2 int coap_server_create (coap_server_t * server, int(*) (coap_server_t *, coap_msg_t *, coap_msg_t *) handle, const char * host, unsigned 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

0	Success
<0	Error

5.7.3.3 void coap_server_destroy (coap_server_t * server)

Deinitialise a server structure.

Parameters

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

5.7.3.4 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

5.7.3.5 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

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

Returns

Operation status

Return values

0	Success
<0	Error

Index

- active
 - coap_server_trans, [13](#)
- COAP_CLIENT_ACK_TIMEOUT_SEC
 - coap_client.c, [31](#)
- COAP_CLIENT_ADDR_BUF_LEN
 - coap_client.h, [15](#)
- COAP_CLIENT_MAX_RETRANSMIT
 - coap_client.c, [31](#)
- COAP_CLIENT_RESP_TIMEOUT_SEC
 - coap_client.c, [31](#)
- COAP_LOG_DEBUG
 - coap_log.h, [17](#)
- COAP_LOG_DEF_LEVEL
 - coap_log.h, [17](#)
- COAP_LOG_ERROR
 - coap_log.h, [17](#)
- COAP_MSG_MAX_BUF_LEN
 - coap_msg.h, [22](#)
- COAP_MSG_MAX_CODE_CLASS
 - coap_msg.h, [22](#)
- COAP_MSG_MAX_CODE_DETAIL
 - coap_msg.h, [22](#)
- COAP_MSG_MAX_MSG_ID
 - coap_msg.h, [22](#)
- COAP_MSG_MAX_TOKEN_LEN
 - coap_msg.h, [22](#)
- COAP_MSG_OP_URI_PATH_MAX_LEN
 - coap_msg.h, [23](#)
- COAP_MSG_OP_URI_PATH_NUM
 - coap_msg.h, [23](#)
- COAP_MSG_VER
 - coap_msg.h, [23](#)
- COAP_SERVER_ACK_TIMEOUT_SEC
 - coap_server.c, [39](#)
- COAP_SERVER_ADDR_BUF_LEN
 - coap_server.h, [29](#)
- COAP_SERVER_MAX_RETRANSMIT
 - coap_server.c, [39](#)
- COAP_SERVER_NUM_TRANS
 - coap_server.h, [29](#)
- client_addr
 - coap_server_trans, [13](#)
- client_sin
 - coap_server_trans, [13](#)
- client_sin_len
 - coap_server_trans, [13](#)
- coap_client.c
 - COAP_CLIENT_ACK_TIMEOUT_SEC, [31](#)
 - COAP_CLIENT_MAX_RETRANSMIT, [31](#)
 - COAP_CLIENT_RESP_TIMEOUT_SEC, [31](#)
 - coap_client_create, [32](#)
 - coap_client_destroy, [32](#)
 - coap_client_exchange, [32](#)
- coap_client.h
 - COAP_CLIENT_ADDR_BUF_LEN, [15](#)
 - coap_client_create, [16](#)
 - coap_client_destroy, [16](#)
 - coap_client_exchange, [16](#)
- coap_client_create
 - coap_client.c, [32](#)
 - coap_client.h, [16](#)
- coap_client_destroy
 - coap_client.c, [32](#)
 - coap_client.h, [16](#)
- coap_client_exchange
 - coap_client.c, [32](#)
 - coap_client.h, [16](#)
- coap_client_t, [7](#)
 - num_retrans, [7](#)
 - sd, [7](#)
 - server_addr, [7](#)
 - server_sin, [7](#)
 - server_sin_len, [7](#)
 - timeout, [8](#)
 - timer_fd, [8](#)
- coap_log.h
 - COAP_LOG_DEBUG, [17](#)
 - COAP_LOG_DEF_LEVEL, [17](#)
 - COAP_LOG_ERROR, [17](#)
 - coap_log_debug, [17](#)
 - coap_log_error, [18](#)
 - coap_log_get_level, [18](#)
 - coap_log_info, [18](#)
 - coap_log_level_t, [17](#)
 - coap_log_notice, [18](#)
 - coap_log_set_level, [18](#)
 - coap_log_warn, [18](#)
- coap_log_debug
 - coap_log.h, [17](#)
- coap_log_error
 - coap_log.h, [18](#)
- coap_log_get_level
 - coap_log.h, [18](#)
- coap_log_info
 - coap_log.h, [18](#)
- coap_log_level_t
 - coap_log.h, [17](#)
- coap_log_notice

- coap_log.h, 18
- coap_log_set_level
 - coap_log.h, 18
- coap_log_warn
 - coap_log.h, 18
- coap_msg.c
 - coap_msg_add_op, 34
 - coap_msg_copy, 34
 - coap_msg_create, 35
 - coap_msg_destroy, 35
 - coap_msg_format, 35
 - coap_msg_gen_rand_str, 35
 - coap_msg_op_list_get_first, 34
 - coap_msg_op_list_get_last, 34
 - coap_msg_op_list_is_empty, 34
 - coap_msg_parse, 35
 - coap_msg_parse_type_msg_id, 36
 - coap_msg_reset, 36
 - coap_msg_set_code, 36
 - coap_msg_set_msg_id, 37
 - coap_msg_set_payload, 37
 - coap_msg_set_token, 37
 - coap_msg_set_type, 38
- coap_msg.h
 - COAP_MSG_MAX_BUF_LEN, 22
 - COAP_MSG_MAX_CODE_CLASS, 22
 - COAP_MSG_MAX_CODE_DETAIL, 22
 - COAP_MSG_MAX_MSG_ID, 22
 - COAP_MSG_MAX_TOKEN_LEN, 22
 - COAP_MSG_OP_URI_PATH_MAX_LEN, 23
 - COAP_MSG_OP_URI_PATH_NUM, 23
 - COAP_MSG_VER, 23
 - coap_msg_add_op, 23
 - coap_msg_copy, 24
 - coap_msg_create, 24
 - coap_msg_destroy, 24
 - coap_msg_format, 24
 - coap_msg_gen_rand_str, 25
 - coap_msg_get_code_class, 21
 - coap_msg_get_code_detail, 21
 - coap_msg_get_first_op, 21
 - coap_msg_get_msg_id, 21
 - coap_msg_get_payload, 21
 - coap_msg_get_payload_len, 21
 - coap_msg_get_token, 21
 - coap_msg_get_token_len, 21
 - coap_msg_get_type, 21
 - coap_msg_get_ver, 22
 - coap_msg_is_empty, 22
 - coap_msg_op_get_len, 22
 - coap_msg_op_get_next, 22
 - coap_msg_op_get_num, 22
 - coap_msg_op_get_val, 22
 - coap_msg_op_num_is_critical, 22
 - coap_msg_op_num_is_unsafe, 23
 - coap_msg_op_num_no_cache_key, 23
 - coap_msg_op_set_len, 23
 - coap_msg_op_set_next, 23
 - coap_msg_op_set_num, 23
 - coap_msg_op_set_val, 23
 - coap_msg_parse, 25
 - coap_msg_parse_type_msg_id, 25
 - coap_msg_reset, 26
 - coap_msg_set_code, 26
 - coap_msg_set_msg_id, 26
 - coap_msg_set_payload, 27
 - coap_msg_set_token, 27
 - coap_msg_set_type, 27
 - coap_msg_add_op
 - coap_msg.c, 34
 - coap_msg.h, 23
 - coap_msg_copy
 - coap_msg.c, 34
 - coap_msg.h, 24
 - coap_msg_create
 - coap_msg.c, 35
 - coap_msg.h, 24
 - coap_msg_destroy
 - coap_msg.c, 35
 - coap_msg.h, 24
 - coap_msg_format
 - coap_msg.c, 35
 - coap_msg.h, 24
 - coap_msg_gen_rand_str
 - coap_msg.c, 35
 - coap_msg.h, 25
 - coap_msg_get_code_class
 - coap_msg.h, 21
 - coap_msg_get_code_detail
 - coap_msg.h, 21
 - coap_msg_get_first_op
 - coap_msg.h, 21
 - coap_msg_get_msg_id
 - coap_msg.h, 21
 - coap_msg_get_payload
 - coap_msg.h, 21
 - coap_msg_get_payload_len
 - coap_msg.h, 21
 - coap_msg_get_token
 - coap_msg.h, 21
 - coap_msg_get_token_len
 - coap_msg.h, 21
 - coap_msg_get_type
 - coap_msg.h, 21
 - coap_msg_get_ver
 - coap_msg.h, 22
 - coap_msg_is_empty
 - coap_msg.h, 22
 - coap_msg_op, 8
 - len, 8
 - next, 8
 - num, 8
 - val, 8
 - coap_msg_op_get_len
 - coap_msg.h, 22
 - coap_msg_op_get_next

- coap_msg.h, [22](#)
- coap_msg_op_get_num
 - coap_msg.h, [22](#)
- coap_msg_op_get_val
 - coap_msg.h, [22](#)
- coap_msg_op_list_get_first
 - coap_msg.c, [34](#)
- coap_msg_op_list_get_last
 - coap_msg.c, [34](#)
- coap_msg_op_list_is_empty
 - coap_msg.c, [34](#)
- coap_msg_op_list_t, [9](#)
 - first, [9](#)
 - last, [9](#)
- coap_msg_op_num_is_critical
 - coap_msg.h, [22](#)
- coap_msg_op_num_is_unsafe
 - coap_msg.h, [23](#)
- coap_msg_op_num_no_cache_key
 - coap_msg.h, [23](#)
- coap_msg_op_set_len
 - coap_msg.h, [23](#)
- coap_msg_op_set_next
 - coap_msg.h, [23](#)
- coap_msg_op_set_num
 - coap_msg.h, [23](#)
- coap_msg_op_set_val
 - coap_msg.h, [23](#)
- coap_msg_parse
 - coap_msg.c, [35](#)
 - coap_msg.h, [25](#)
- coap_msg_parse_type_msg_id
 - coap_msg.c, [36](#)
 - coap_msg.h, [25](#)
- coap_msg_reset
 - coap_msg.c, [36](#)
 - coap_msg.h, [26](#)
- coap_msg_set_code
 - coap_msg.c, [36](#)
 - coap_msg.h, [26](#)
- coap_msg_set_msg_id
 - coap_msg.c, [37](#)
 - coap_msg.h, [26](#)
- coap_msg_set_payload
 - coap_msg.c, [37](#)
 - coap_msg.h, [27](#)
- coap_msg_set_token
 - coap_msg.c, [37](#)
 - coap_msg.h, [27](#)
- coap_msg_set_type
 - coap_msg.c, [38](#)
 - coap_msg.h, [27](#)
- coap_msg_t, [9](#)
 - code_class, [10](#)
 - code_detail, [10](#)
 - msg_id, [10](#)
 - op_list, [10](#)
 - payload, [10](#)
 - payload_len, [10](#)
 - token, [10](#)
 - token_len, [10](#)
 - type, [10](#)
 - ver, [10](#)
- coap_server, [11](#)
 - handle, [11](#)
 - msg_id, [11](#)
 - sd, [11](#)
 - sep_list, [11](#)
 - trans, [11](#)
- coap_server.c
 - COAP_SERVER_ACK_TIMEOUT_SEC, [39](#)
 - COAP_SERVER_MAX_RETRANSMIT, [39](#)
 - coap_server_add_sep_resp_uri_path, [39](#)
 - coap_server_create, [39](#)
 - coap_server_destroy, [40](#)
 - coap_server_get_next_msg_id, [40](#)
 - coap_server_run, [40](#)
- coap_server.h
 - COAP_SERVER_ADDR_BUF_LEN, [29](#)
 - COAP_SERVER_NUM_TRANS, [29](#)
 - coap_server_add_sep_resp_uri_path, [29](#)
 - coap_server_create, [29](#)
 - coap_server_destroy, [30](#)
 - coap_server_get_next_msg_id, [30](#)
 - coap_server_run, [30](#)
- coap_server_add_sep_resp_uri_path
 - coap_server.c, [39](#)
 - coap_server.h, [29](#)
- coap_server_create
 - coap_server.c, [39](#)
 - coap_server.h, [29](#)
- coap_server_destroy
 - coap_server.c, [40](#)
 - coap_server.h, [30](#)
- coap_server_get_next_msg_id
 - coap_server.c, [40](#)
 - coap_server.h, [30](#)
- coap_server_path, [12](#)
 - next, [12](#)
 - str, [12](#)
- coap_server_path_list_t, [12](#)
 - first, [12](#)
 - last, [12](#)
- coap_server_run
 - coap_server.c, [40](#)
 - coap_server.h, [30](#)
- coap_server_trans, [13](#)
 - active, [13](#)
 - client_addr, [13](#)
 - client_sin, [13](#)
 - client_sin_len, [13](#)
 - last_use, [13](#)
 - num_retrans, [14](#)
 - req, [14](#)
 - resp, [14](#)
 - server, [14](#)

- timeout, [14](#)
 - timer_fd, [14](#)
- code_class
 - coap_msg_t, [10](#)
- code_detail
 - coap_msg_t, [10](#)
- first
 - coap_msg_op_list_t, [9](#)
 - coap_server_path_list_t, [12](#)
- handle
 - coap_server, [11](#)
- include/coap_client.h, [15](#)
- include/coap_log.h, [16](#)
- include/coap_msg.h, [19](#)
- include/coap_server.h, [28](#)
- last
 - coap_msg_op_list_t, [9](#)
 - coap_server_path_list_t, [12](#)
- last_use
 - coap_server_trans, [13](#)
- len
 - coap_msg_op, [8](#)
- msg_id
 - coap_msg_t, [10](#)
 - coap_server, [11](#)
- next
 - coap_msg_op, [8](#)
 - coap_server_path, [12](#)
- num
 - coap_msg_op, [8](#)
- num_retrans
 - coap_client_t, [7](#)
 - coap_server_trans, [14](#)
- op_list
 - coap_msg_t, [10](#)
- payload
 - coap_msg_t, [10](#)
- payload_len
 - coap_msg_t, [10](#)
- req
 - coap_server_trans, [14](#)
- resp
 - coap_server_trans, [14](#)
- sd
 - coap_client_t, [7](#)
 - coap_server, [11](#)
- sep_list
 - coap_server, [11](#)
- server
 - coap_server_trans, [14](#)
- server_addr
 - coap_client_t, [7](#)
- server_sin
 - coap_client_t, [7](#)
- server_sin_len
 - coap_client_t, [7](#)
- src/coap_client.c, [31](#)
- src/coap_msg.c, [32](#)
- src/coap_server.c, [38](#)
- str
 - coap_server_path, [12](#)
- timeout
 - coap_client_t, [8](#)
 - coap_server_trans, [14](#)
- timer_fd
 - coap_client_t, [8](#)
 - coap_server_trans, [14](#)
- token
 - coap_msg_t, [10](#)
- token_len
 - coap_msg_t, [10](#)
- trans
 - coap_server, [11](#)
- type
 - coap_msg_t, [10](#)
- val
 - coap_msg_op, [8](#)
- ver
 - coap_msg_t, [10](#)