FreeCoAP

Generated by Doxygen 1.8.10

Wed Sep 2 2015 19:30:59

Contents

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

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:

Chapter 3

File Index

3.1 File List

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

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 structture

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	*client*	Pointer to a client structure
in	*host*	Pointer to a string containing the host address of the server
in	*port*	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	*client*	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	*client*	Pointer to a client structure
in	*req*	Pointer to the request message
out	*resp*	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	*msg*	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	*msg*	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	*msg*	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	*msg*	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	*level*	The new log level

5.2.4.7 void coap_log_warn (const char ∗ *msg*, ...)

Log a warning message.

Parameters

in	*msg*	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, **CO↩ AP_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_CHA↩ NGED** = 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, **CO↩ AP_MSG_FORBIDDEN** = 3,
  COAP_MSG_NOT_FOUND = 4, **COAP_MSG_METHOD_NOT_ALLOWED** = 5, **COAP_MSG_NOT_AC↩ CEPTABLE** = 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

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

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.2 int coap_msg_copy (coap_msg_t ∗ *dst,* coap_msg_t ∗ *src*)

Copy a message.

Parameters

in,out	*dst*	Pointer to the destination message structure
in	*src*	Pointer to the source message structure

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.3 void coap_msg_create (coap_msg_t ∗ *msg*)

Initialise a message structure.

Parameters

out	*msg*	Pointer to a message structure

5.3.3.4 void coap_msg_destroy (coap_msg_t ∗ *msg*)

Deinitialise a message structure.

Parameters

in,out	*msg*	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	*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

5.3.3.6 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

5.3.3.7 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

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	*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

5.3.3.9 void coap_msg_reset (coap_msg_t ∗ *msg*)

Deinitialise and initialise a message structure.

Parameters

in,out	*msg*	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	*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

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	*msg*	Pointer to a message structure
in	*msg_id*	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	*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
in	*port*	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	*server*	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	*server*	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	*server*	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	*client*	Pointer to a client structure
in	*host*	Pointer to a string containing the host address of the server
in	*port*	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	*client*	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	*client*	Pointer to a client structure
in	*req*	Pointer to the request message
out	*resp*	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

`in,out`	*msg*	Pointer to a message structure
`in`	*num*	Option number
`in`	*len*	Option length
`in`	*val*	Pointer to a buffer containing the option value

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.2 int coap_msg_copy (coap_msg_t * *dst,* coap_msg_t * *src*)

Copy a message.

Parameters

`in,out`	*dst*	Pointer to the destination message structure
`in`	*src*	Pointer to the source message structure

Returns

Operation status

Return values

0	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

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

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

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

5.6.3.9 void coap_msg_reset (coap_msg_t ∗ *msg*)

Deinitialise and initialise a message structure.

Parameters

in,out	*msg*	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

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

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	*msg*	Pointer to a message structure
in	*msg_id*	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	*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

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

Set the token in a message.

Parameters

out	*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

5.6.3.14 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

5.7 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`	*server*	Pointer to a server structure
`in`	*str*	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	*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
in	*port*	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	*server*	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	*server*	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	*server*	Pointer to a server structure

Returns

Operation status

Return values

0	Success
<0	Error

Index