

# libosip Reference Manual

## 0.8.5

Generated by Doxygen 1.2.15

Fri May 17 12:14:37 2002



---

# Contents

<b>1</b>	<b>libosip Module Index</b>	<b>1</b>
1.1	libosip Modules . . . . .	1
<b>2</b>	<b>libosip File Index</b>	<b>3</b>
2.1	libosip File List . . . . .	3
<b>3</b>	<b>libosip Module Documentation</b>	<b>5</b>
3.1	oSIP dialog Handling . . . . .	5
3.2	oSIP fifo Handling . . . . .	8
3.3	oSIP list Handling . . . . .	10
3.4	oSIP fsm Handling . . . . .	12
3.5	oSIP SDP parser Handling . . . . .	39
3.6	oSIP and SDP offer/answer model Handling . . . . .	54
3.7	oSIP semaphore definitions . . . . .	63
3.8	oSIP parser Handling . . . . .	65
3.9	oSIP type definitions . . . . .	155
3.10	oSIP url parser Handling . . . . .	160
<b>4</b>	<b>libosip File Documentation</b>	<b>171</b>
4.1	dialog.h File Reference . . . . .	171
4.2	fifo.h File Reference . . . . .	172
4.3	list.h File Reference . . . . .	173
4.4	osip.h File Reference . . . . .	174
4.5	sdp.h File Reference . . . . .	178
4.6	sdp_negoc.h File Reference . . . . .	180
4.7	sema.h File Reference . . . . .	182
4.8	smsg.h File Reference . . . . .	183
4.9	smsgtypes.h File Reference . . . . .	195
4.10	urls.h File Reference . . . . .	197

---



---

# Chapter 1

# libosip Module Index

## 1.1 libosip Modules

Here is a list of all modules:

oSIP dialog Handling . . . . .	5
oSIP fifo Handling . . . . .	8
oSIP list Handling . . . . .	10
oSIP fsm Handling . . . . .	12
oSIP SDP parser Handling . . . . .	39
oSIP and SDP offer/answer model Handling . . . . .	54
oSIP semaphore definitions . . . . .	63
oSIP parser Handling . . . . .	65
oSIP type definitions . . . . .	155
oSIP Thread Routines . . . . .	??
oSIP url parser Handling . . . . .	160



---

## Chapter 2

# libosip Compound Index

### 2.1 libosip Compound List

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

**dialog\_t** . . . . . ??





---

## Chapter 3

# libosip File Index

### 3.1 libosip File List

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

<b>const.h</b>	??
<b>dialog.h</b> (OSIP dialog Routines)	171
<b>fifo.h</b> (OSIP fifo Routines)	172
<b>fsm.h</b>	??
<b>global.h</b>	??
<b>list.h</b> (OSIP list Routines)	173
<b>md5.h</b>	??
<b>msg.h</b>	??
<b>osip.h</b> (OSIP fsm Routines)	174
<b>port.h</b>	??
<b>sdp.h</b> (OSIP SDP parser Routines)	178
<b>sdp_negoc.h</b> (OSIP and SDP offer/answer model Routines)	180
<b>sema.h</b> (OSIP semaphore definitions)	182
<b>smsg.h</b> (OSIP parser Routines)	183
<b>smsgtypes.h</b> (OSIP type definitions)	195
<b>thread.h</b> (OSIP Thread Routines)	??
<b>urls.h</b> (OSIP url parser Routines)	197



---

## Chapter 4

# libosip Module Documentation

### 4.1 oSIP dialog Handling

#### Compounds

- struct **dialog\_t**

#### Typedefs

- typedef dialog\_t **dialog\_t**

#### Functions

- int **dialog\_init\_as\_uac** (dialog\_t \*\*dialog, sip\_t \*response)
- int **dialog\_init\_as\_uas** (dialog\_t \*\*dialog, sip\_t \*invite, sip\_t \*response)
- void **dialog\_free** (dialog\_t \*dialog)
- void **dialog\_set\_state** (dialog\_t \*dialog, dlg\_type\_t type)
- int **dialog\_update\_route\_set\_as\_uas** (dialog\_t \*dialog, sip\_t \*invite)
- int **dialog\_update\_cseq\_as\_uas** (dialog\_t \*dialog, sip\_t \*request)
- int **dialog\_match\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_update\_tag\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_update\_route\_set\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_match\_as\_uas** (dialog\_t \*dialog, sip\_t \*request)

#### 4.1.1 Typedef Documentation

##### 4.1.1.1 dialog\_t

Structure for referencing a dialog.

#### 4.1.2 Function Documentation

##### 4.1.2.1 void dialog\_free (dialog\_t \* *dialog*)

Free all resource in a **dialog\_t** (p.??) element.

---

**Parameters:**

*dialog* The element to free.

**4.1.2.2 int dialog\_init\_as\_uac (dialog\_t \*\* *dialog*, sip\_t \* *response*)**

Allocate a **dialog\_t** (p.??) element as a UAC.

- NOTE1: Only INVITE transactions can create a dialog.
- NOTE2: The dialog should be created when the first response is received. (except for a 100 Trying)
- NOTE3: Remote UA should be compliant! If not (not tag in the to header?) the old mechanism is used to match the request but if 2 uncompliant UA both answer 200 OK for the same transaction, they won't be detected. This is a major BUG in the old rfc.

**Parameters:**

*dialog* The element to allocate.

*response* The response containing the informations.

**4.1.2.3 int dialog\_init\_as\_uas (dialog\_t \*\* *dialog*, sip\_t \* *invite*, sip\_t \* *response*)**

Allocate a **dialog\_t** (p.??) element as a UAS. NOTE1: Only INVITE transactions can create a dialog. NOTE2: The dialog should be created when the first response is sent. (except for a 100 Trying)

**Parameters:**

*dialog* The element to allocate.

*invite* The INVITE request containing some informations.

*response* The response containing other informations.

**4.1.2.4 int dialog\_match\_as\_uac (dialog\_t \* *dialog*, sip\_t \* *response*)**

Match a response received with a dialog.

**Parameters:**

*dialog* The element to work on.

*response* The response received.

**4.1.2.5 int dialog\_match\_as\_uas (dialog\_t \* *dialog*, sip\_t \* *request*)**

Match a request (response sent??) received with a dialog.

**Parameters:**

*dialog* The element to work on.

*request* The request received.

**4.1.2.6 void dialog\_set\_state (dialog\_t \* *dialog*, dlg\_type\_t *type*)**

Set the state of the dialog. This is useful to keep information on who is the initiator of the call.

**Parameters:**

*dialog* The element to work on.

*type* The type of dialog (CALLEE or CALLER).

**4.1.2.7 int dialog\_update\_cseq\_as\_uas (dialog\_t \* *dialog*, sip\_t \* *request*)**

Update the CSeq (remote cseq) during a UAS transaction of a dialog. NOTE: All INCOMING transactions MUST update the remote CSeq.

**Parameters:**

*dialog* The element to work on.

*request* The request received.

**4.1.2.8 int dialog\_update\_route\_set\_as\_uac (dialog\_t \* *dialog*, sip\_t \* *response*)**

Update the Route-Set as UAC of a dialog. NOTE: bis-09 says that only INVITE transactions can update the route-set. NOTE: bis-09 says that updating the route-set means: update the contact field only (AND NOT THE ROUTE-SET). This method follow this behaviour. NOTE: This method should be called for each request (except 100 Trying) received for a dialog.

**Parameters:**

*dialog* The element to work on.

*response* The response received.

**4.1.2.9 int dialog\_update\_route\_set\_as\_uas (dialog\_t \* *dialog*, sip\_t \* *invite*)**

Update the Route-Set as UAS of a dialog. NOTE: bis-09 says that only INVITE transactions can update the route-set. NOTE: bis-09 says that updating the route-set means: update the contact field only (AND NOT THE ROUTE-SET). This method follow this behaviour. NOTE: This method should be called for each request (except 100 Trying) received for a dialog.

**Parameters:**

*dialog* The element to work on.

*invite* The invite received.

**4.1.2.10 int dialog\_update\_tag\_as\_uac (dialog\_t \* *dialog*, sip\_t \* *response*)**

Update the tag as UAC of a dialog?. (this could be needed if the 180 does not contains any tag, but the 200 contains one.

**Parameters:**

*dialog* The element to work on.

*response* The response received.

## 4.2 oSIP fifo Handling

### Compounds

- struct **fifo\_t**

### Typedefs

- typedef fifo\_t **fifo\_t**

### Functions

- void **fifo\_init** (fifo\_t \*ff)
- void **fifo\_free** (fifo\_t \*ff)
- int **fifo\_add** (fifo\_t \*ff, void \*element)
- void \* **fifo\_get** (fifo\_t \*ff)
- void \* **fifo\_tryget** (fifo\_t \*ff)

#### 4.2.1 Typedef Documentation

##### 4.2.1.1 typedef struct fifo\_t fifo\_t

Structure for referencing a fifo. @defvar fifo\_t

#### 4.2.2 Function Documentation

##### 4.2.2.1 int fifo\_add (fifo\_t \* *ff*, void \* *element*)

Add an element in a fifo.

**Parameters:**

*ff* The element to work on.

*element* The pointer on the element to add.

##### 4.2.2.2 void fifo\_free (fifo\_t \* *ff*)

Free a fifo element.

**Parameters:**

*ff* The element to work on.

##### 4.2.2.3 void\* fifo\_get (fifo\_t \* *ff*)

Get an element from a fifo or block until one is added.

**Parameters:**

*ff* The element to work on.

**4.2.2.4 void fifo\_init (fifo\_t \* *ff*)**

Initialise a fifo\_t element. NOTE: this element MUST be previously allocated.

**Parameters:**

*ff* The element to initialise.

**4.2.2.5 void\* fifo\_tryget (fifo\_t \* *ff*)**

Try to get an element from a fifo, but do not block if there is no element.

**Parameters:**

*ff* The element to work on.

## 4.3 oSIP list Handling

### Compounds

- struct `list_t`

### Typedefs

- typedef `list_t list_t`

### Functions

- int `list_init` (`list_t *li`)
- void `list_special_free` (`list_t *li`, void `*(free_func)(void *)`)
- void `listofchar_free` (`list_t *li`)
- int `list_size` (`list_t *li`)
- int `list_eol` (`list_t *li`, int `pos`)
- int `list_add` (`list_t *li`, void `*element`, int `pos`)
- void `*list_get` (`list_t *li`, int `pos`)
- int `list_remove` (`list_t *li`, int `pos`)

#### 4.3.1 Typedef Documentation

##### 4.3.1.1 typedef struct `list_t list_t`

Structure for referencing a list of elements. @defvar `list_t`

#### 4.3.2 Function Documentation

##### 4.3.2.1 int `list_add` (`list_t *li`, void `*element`, int `pos`)

Add an element in a list.

**Parameters:**

*li* The element to work on.

*element* The pointer on the element to add.

*pos* the index of the element to add. (or -1 to append the element at the end)

##### 4.3.2.2 int `list_eol` (`list_t *li`, int `pos`)

Check if the end of list is detected .

**Parameters:**

*li* The element to work on.

*pos* The index of the possible element.



**4.3.2.3 void\* list\_get (list\_t \* *li*, int *pos*)**

Get an element from a list.

**Parameters:**

- li* The element to work on.
- pos* the index of the element to get.

**4.3.2.4 int list\_init (list\_t \* *li*)**

Initialise a list\_t element. NOTE: this element MUST be previously allocated.

**Parameters:**

- li* The element to initialise.

**4.3.2.5 int list\_remove (list\_t \* *li*, int *pos*)**

Remove an element from a list.

**Parameters:**

- li* The element to work on.
- pos* the index of the element to remove.

**4.3.2.6 int list\_size (list\_t \* *li*)**

Get the size of a list of element.

**Parameters:**

- li* The element to work on.

**4.3.2.7 void list\_special\_free (list\_t \* *li*, void \*(\* *free\_func*)(void \*))**

Free a list of element. Each element will be free with the method given as the second parameter.

**Parameters:**

- li* The element to work on.
- free\_func* The method that is able to release one element of the list.

**4.3.2.8 void listofchar\_free (list\_t \* *li*)**

Free a list of element where elements are pointer to 'char'.

**Parameters:**

- li* The element to work on.

## 4.4 oSIP fsm Handling

### Compounds

- struct `ict_t`
- struct `ist_t`
- struct `nict_t`
- struct `nist_t`
- struct `osip_t`
- struct `sipevent_t`
- struct `transaction_t`

### Defines

- #define `SIP_MESSAGE_MAX_LENGTH` 4000
- #define `DEFAULT_T1` 500
- #define `DEFAULT_T2` 4000
- #define `DEFAULT_T4` 5000
- #define `EVT_IS_RCV_INVITE`(event) (event → type==RCV\_REQINVITE)
- #define `EVT_IS_RCV_ACK`(event) (event → type==RCV\_REQACK)
- #define `EVT_IS_RCV_REQUEST`(event) (event → type==RCV\_REQUEST)
- #define `EVT_IS_RCV_STATUS_1XX`(event) (event → type==RCV\_STATUS\_1XX)
- #define `EVT_IS_RCV_STATUS_2XX`(event) (event → type==RCV\_STATUS\_2XX)
- #define `EVT_IS_RCV_STATUS_3456XX`(event) (event → type==RCV\_STATUS\_3456XX)
- #define `EVT_IS_SND_INVITE`(event) (event → type==SND\_REQINVITE)
- #define `EVT_IS_SND_ACK`(event) (event → type==SND\_REQACK)
- #define `EVT_IS_SND_REQUEST`(event) (event → type==SND\_REQUEST)
- #define `EVT_IS_SND_STATUS_1XX`(event) (event → type==SND\_STATUS\_1XX)
- #define `EVT_IS_SND_STATUS_2XX`(event) (event → type==SND\_STATUS\_2XX)
- #define `EVT_IS_SND_STATUS_3456XX`(event) (event → type==SND\_STATUS\_3456XX)
- #define `EVT_IS_INCOMINGMSG`(event)
- #define `EVT_IS_INCOMINGREQ`(event)
- #define `EVT_IS_INCOMINGRESP`(event)
- #define `EVT_IS_OUTGOINGMSG`(event)
- #define `EVT_IS_OUTGOINGREQ`(event)
- #define `EVT_IS_OUTGOINGRESP`(event)
- #define `EVT_IS_MSG`(event)
- #define `EVT_IS_KILL_TRANSACTION`(event) (event → type==KILL\_TRANSACTION)

### Typedefs

- typedef enum `_state_t` `state_t`
- typedef enum `type_t` `type_t`
- typedef enum `context_type_t` `context_type_t`
- typedef `ict_t` `ict_t`
- typedef `nict_t` `nict_t`
- typedef `ist_t` `ist_t`

- typedef nist\_t **nist\_t**
- typedef transaction\_t **transaction\_t**
- typedef osip\_t **osip\_t**
- typedef sipevent\_t **sipevent\_t**

## Enumerations

- enum **\_state\_t** { **ICT\_PRE\_CALLING**, **ICT\_CALLING**, **ICT\_PROCEEDING**, **ICT\_COMPLETED**, **ICT\_TERMINATED**, **IST\_PRE\_PROCEEDING**, **IST\_PROCEEDING**, **IST\_COMPLETED**, **IST\_CONFIRMED**, **IST\_TERMINATED**, **NICT\_PRE\_TRYING**, **NICT\_TRYING**, **NICT\_PROCEEDING**, **NICT\_COMPLETED**, **NICT\_TERMINATED**, **NIST\_PRE\_TRYING**, **NIST\_TRYING**, **NIST\_PROCEEDING**, **NIST\_COMPLETED**, **NIST\_TERMINATED** }

## Functions

- int **ict\_set\_destination** (ict\_t \*ict, char \*destination, int port)
- int **nict\_set\_destination** (nict\_t \*nict, char \*destination, int port)
- sipevent\_t \* **nist\_need\_timer\_j\_event** (nist\_t \*nist, state\_t state, int transactionid)
- int **transaction\_init** (transaction\_t \*\*transaction, context\_type\_t ctx\_type, osip\_t \*osip, sip\_t \*request)
- int **transaction\_free** (transaction\_t \*transaction)
- int **transaction\_add\_event** (transaction\_t \*transaction, sipevent\_t \*evt)
- int **transaction\_execute** (transaction\_t \*transaction, sipevent\_t \*evt)
- int **transaction\_set\_your\_instance** (transaction\_t \*transaction, void \*instance)
- void \* **transaction\_get\_your\_instance** (transaction\_t \*transaction)
- int **osip\_global\_init** ()
- void **osip\_global\_free** ()
- int **osip\_init** (osip\_t \*\*osip)
- void **osip\_free** (osip\_t \*osip)
- int **osip\_ict\_execute** (osip\_t \*osip)
- int **osip\_ist\_execute** (osip\_t \*osip)
- int **osip\_nict\_execute** (osip\_t \*osip)
- int **osip\_nist\_execute** (osip\_t \*osip)
- void **osip\_timers\_ict\_execute** (osip\_t \*osip)
- void **osip\_timers\_ist\_execute** (osip\_t \*osip)
- void **osip\_timers\_nict\_execute** (osip\_t \*osip)
- void **osip\_timers\_nist\_execute** (osip\_t \*osip)
- transaction\_t \* **osip\_transaction\_find** (list\_t \*transactions, sipevent\_t \*evt)
- transaction\_t \* **osip\_find\_transaction** (osip\_t \*osip, sipevent\_t \*evt)
- transaction\_t \* **osip\_create\_transaction** (osip\_t \*osip, sipevent\_t \*evt)
- sipevent\_t \* **osip\_parse** (char \*buf)
- sipevent\_t \* **osip\_new\_outgoing\_sipmessage** (sip\_t \*sip)
- void **osip\_setcb\_send\_message** (osip\_t \*cf, int(\*cb)(transaction\_t \*, sip\_t \*, char \*, int, int))
- void **osip\_setcb\_ict\_kill\_transaction** (osip\_t \*cf, void(\*cb)(transaction\_t \*))
- void **osip\_setcb\_ict\_invite\_sent** (osip\_t \*cf, void(\*cb)(transaction\_t \*, sip\_t \*))
- void **osip\_setcb\_ict\_invite\_sent2** (osip\_t \*cf, void(\*cb)(transaction\_t \*, sip\_t \*))
- void **osip\_setcb\_ict\_ack\_sent** (osip\_t \*cf, void(\*cb)(transaction\_t \*, sip\_t \*))
- void **osip\_setcb\_ict\_ack\_sent2** (osip\_t \*cf, void(\*cb)(transaction\_t \*, sip\_t \*))

- void `osip_setcb_ict_1xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_2xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_2xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_3xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_4xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_5xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_6xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_3456xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_transport_error` (`osip_t *cf`, `void(*cb)(transaction_t *, int error)`)
- void `osip_setcb_ist_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_ist_invite_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_invite_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_ack_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_ack_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_1xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_1xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_2xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_2xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_3xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_4xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_5xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_6xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_3456xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_transport_error` (`osip_t *cf`, `void(*cb)(transaction_t *, int error)`)
- void `osip_setcb_nict_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_nict_register_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_bye_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_options_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_info_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_cancel_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_notify_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_subscribe_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_unknown_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_request_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_1xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_2xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_2xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_3xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_4xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_5xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_6xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_3456xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_transport_error` (`osip_t *cf`, `void(*cb)(transaction_t *, int error)`)
- void `osip_setcb_nist_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_nist_register_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nist_bye_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nist_options_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nist_info_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nist_cancel_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nist_notify_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)

- void `osip_setcb_nist_subscribe_received` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_unknown_received` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_request_received2` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_1xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_2xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_2xx_sent2` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_3xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_4xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_5xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_6xx_sent` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_3456xx_sent2` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_transport_error` (`osip_t *cf`, void(`*cb`)(`transaction_t *`, int error))

#### 4.4.1 Define Documentation

##### 4.4.1.1 `#define DEFAULT_T1 500`

You can re-define the default value for T1. (T1 is defined in rfcxxxx) The default value is 500ms.

##### 4.4.1.2 `#define DEFAULT_T2 4000`

You can re-define the default value for T2. (T2 is defined in rfcxxxx) The default value is 4000ms.

##### 4.4.1.3 `#define DEFAULT_T4 5000`

You can re-define the default value for T4. (T1 is defined in rfcxxxx) The default value is 5000ms.

##### 4.4.1.4 `#define EVT_IS_INCOMINGMSG(event)`

**Value:**

```
(event->type>=RCV_REQINVITE \
                                &&event->type<=RCV_STATUS_3456XX)
```

Check if the sipevent is of an incoming SIP MESSAGE.

**Parameters:**

*event* the event to check.

##### 4.4.1.5 `#define EVT_IS_INCOMINGREQ(event)`

**Value:**

```
(EVT_IS_RCV_INVITE(event) \
||EVT_IS_RCV_ACK(event) \
||EVT_IS_RCV_REQUEST(event))
```

Check if the sipevent is of an incoming SIP REQUEST.

**Parameters:**

*event* the event to check.

#### 4.4.1.6 #define EVT\_IS\_INCOMINGRESP(event)

**Value:**

```
(EVT_IS_RCV_STATUS_1XX(event) \
||EVT_IS_RCV_STATUS_2XX(event) \
||EVT_IS_RCV_STATUS_3456XX(event))
```

Check if the sipevent is of an incoming SIP RESPONSE.

**Parameters:**

*event* the event to check.

#### 4.4.1.7 #define EVT\_IS\_KILL\_TRANSACTION(event) (event → type==KILL\_TRANSACTION)

Check if the sipevent is of type KILL\_TRANSACTION. NOTE: THIS IS AN INTERNAL METHOD ONLY

**Parameters:**

*event* the event to check.

#### 4.4.1.8 #define EVT\_IS\_MSG(event)

**Value:**

```
(event->type>=RCV_REQINVITE \
&&event->type<=SND_STATUS_3456XX)
```

Check if the sipevent is a SIP MESSAGE.

**Parameters:**

*event* the event to check.

#### 4.4.1.9 #define EVT\_IS\_OUTGOINGMSG(event)

**Value:**

```
(event->type>=SND_REQINVITE \
&&event->type<=SND_STATUS_3456XX)
```

Check if the sipevent is of an outgoing SIP MESSAGE.

**Parameters:**

*event* the event to check.

**4.4.1.10 #define EVT\_IS\_OUTGOINGREQ(event)****Value:**

```
(EVT_IS_SND_INVITE(event) \
                                ||EVT_IS_SND_ACK(event) \
                                ||EVT_IS_SND_REQUEST(event))
```

Check if the sipevent is of an outgoing SIP REQUEST.

**Parameters:**

*event* the event to check.

**4.4.1.11 #define EVT\_IS\_OUTGOINGRESP(event)****Value:**

```
(EVT_IS_SND_STATUS_1XX(event) \
                                ||EVT_IS_SND_STATUS_2XX(event) \
                                ||EVT_IS_SND_STATUS_3456XX(event))
```

Check if the sipevent is of an outgoing SIP RESPONSE.

**Parameters:**

*event* the event to check.

**4.4.1.12 #define EVT\_IS\_RCV\_ACK(event) (event → type==RCV\_REQACK)**

Check if the sipevent is of type RCV\_REQACK.

**Parameters:**

*event* the event to check.

**4.4.1.13 #define EVT\_IS\_RCV\_INVITE(event) (event → type==RCV\_REQINVITE)**

Check if the sipevent is of type RCV\_REQINVITE.

**Parameters:**

*event* the event to check.

**4.4.1.14 #define EVT\_IS\_RCV\_REQUEST(event) (event → type==RCV\_REQUEST)**

Check if the sipevent is of type RCV\_REQUEST.

**Parameters:**

*event* the event to check.

**4.4.1.15** `#define EVT_IS_RCV_STATUS_1XX(event) (event → type==RCV_STATUS_1XX)`

Check if the sipevent is of type RCV\_STATUS\_1XX.

**Parameters:**

*event* the event to check.

**4.4.1.16** `#define EVT_IS_RCV_STATUS_2XX(event) (event → type==RCV_STATUS_2XX)`

Check if the sipevent is of type RCV\_STATUS\_2XX.

**Parameters:**

*event* the event to check.

**4.4.1.17** `#define EVT_IS_RCV_STATUS_3456XX(event) (event → type==RCV_STATUS_3456XX)`

Check if the sipevent is of type RCV\_STATUS\_3456XX.

**Parameters:**

*event* the event to check.

**4.4.1.18** `#define EVT_IS_SND_ACK(event) (event → type==SND_REQACK)`

Check if the sipevent is of type SND\_REQACK.

**Parameters:**

*event* the event to check.

**4.4.1.19** `#define EVT_IS_SND_INVITE(event) (event → type==SND_REQINVITE)`

Check if the sipevent is of type SND\_REQINVITE.

**Parameters:**

*event* the event to check.

**4.4.1.20** `#define EVT_IS_SND_REQUEST(event) (event → type==SND_REQUEST)`

Check if the sipevent is of type SND\_REQUEST.

**Parameters:**

*event* the event to check.



**4.4.1.21** `#define EVT_IS_SND_STATUS_1XX(event) (event → type==SND_STATUS_1XX)`

Check if the sipevent is of type SND\_STATUS\_1XX.

**Parameters:**

*event* the event to check.

**4.4.1.22** `#define EVT_IS_SND_STATUS_2XX(event) (event → type==SND_STATUS_2XX)`

Check if the sipevent is of type SND\_STATUS\_2XX.

**Parameters:**

*event* the event to check.

**4.4.1.23** `#define EVT_IS_SND_STATUS_3456XX(event) (event → type==SND_STATUS_3456XX)`

Check if the sipevent is of type SND\_STATUS\_3456XX.

**Parameters:**

*event* the event to check.

**4.4.1.24** `#define SIP_MESSAGE_MAX_LENGTH 4000`

You can re-define your own maximum length for SIP message.

The default value is 20000 characters. (which is much more that the MTU!)

BUG: If you try to build a message string (by calling msg\_2char), with a higher length, your application will crash. Anybody building a commercial application should fix that behavior (and hopefully send the code back to me!)

## 4.4.2 Typedef Documentation

**4.4.2.1** `typedef enum context_type_t context_type_t`

Enumeration for transaction type. A transaction can be either of: ICT, IST, NICT, NIST,

**4.4.2.2** `typedef struct ict_t ict_t`

Structure for INVITE CLIENT TRANSACTION (outgoing INVITE transaction). @defvar ict\_t

**4.4.2.3** `typedef struct ist_t ist_t`

Structure for INVITE SERVER TRANSACTION (incoming INVITE transaction). @defvar ist\_t

#### 4.4.2.4 `typedef struct nict_t nict_t`

Structure for NON-INVITE CLIENT TRANSACTION (outgoing NON-INVITE transaction).  
@defvar nict\_t

#### 4.4.2.5 `typedef struct nist_t nist_t`

Structure for NON-INVITE SERVER TRANSACTION (incoming SERVER transaction). @defvar nist\_t

#### 4.4.2.6 `typedef struct osip_t osip_t`

Structure for osip handling. In order to use osip, you have to manage at least one global instance of an osip\_t element. Then, you'll register a set of required callbacks and a set of optional ones.  
@defvar osip\_t

#### 4.4.2.7 `typedef struct sipevent_t sipevent_t`

Structure for sipevent handling. A sipevent\_t element will have a type and will be related to a transaction. In the general case, it is used by the application layer to give SIP messages to the oSIP finite state machine. @defvar sipevent\_t

#### 4.4.2.8 `typedef enum _state_t state_t`

Enumeration for transaction state.

Here is the list of possible values for transactions:

ICT\_PRE\_CALLING,  
ICT\_CALLING,  
ICT\_PROCEEDING,  
ICT\_COMPLETED,  
ICT\_TERMINATED,  
IST\_PRE\_PROCEEDING,  
IST\_PROCEEDING,  
IST\_COMPLETED,  
IST\_CONFIRMED,  
IST\_TERMINATED,  
NICT\_PRE\_TRYING,  
NICT\_TRYING,  
NICT\_PROCEEDING,  
NICT\_COMPLETED,  
NICT\_TERMINATED,  
NIST\_PRE\_TRYING,  
NIST\_TRYING,

NIST\_PROCEEDING,  
NIST\_COMPLETED,  
NIST\_TERMINATED,

#### 4.4.2.9 typedef struct transaction\_t transaction\_t

Structure for transaction handling. @defvar transaction\_t

#### 4.4.2.10 typedef enum type\_t type\_t

Enumeration for event type.

The list of values that you need to know is reduced to this:

RCV\_REQINVITE,  
RCV\_REQACK,  
RCV\_REQUEST,  
RCV\_STATUS\_1XX,  
RCV\_STATUS\_2XX,  
RCV\_STATUS\_3456XX,  
SND\_REQINVITE,  
SND\_REQACK,  
SND\_REQUEST,  
SND\_STATUS\_1XX,  
SND\_STATUS\_2XX,  
SND\_STATUS\_3456XX,

### 4.4.3 Enumeration Type Documentation

#### 4.4.3.1 enum \_state\_t

Enumeration for transaction state.

Here is the list of possible values for transactions:

ICT\_PRE\_CALLING,  
ICT\_CALLING,  
ICT\_PROCEEDING,  
ICT\_COMPLETED,  
ICT\_TERMINATED,  
IST\_PRE\_PROCEEDING,  
IST\_PROCEEDING,  
IST\_COMPLETED,  
IST\_CONFIRMED,

IST\_TERMINATED,  
 NICT\_PRE\_TRYING,  
 NICT\_TRYING,  
 NICT\_PROCEEDING,  
 NICT\_COMPLETED,  
 NICT\_TERMINATED,  
 NIST\_PRE\_TRYING,  
 NIST\_TRYING,  
 NIST\_PROCEEDING,  
 NIST\_COMPLETED,  
 NIST\_TERMINATED,

#### 4.4.4 Function Documentation

##### 4.4.4.1 int ict\_set\_destination (ict\_t \* *ict*, char \* *destination*, int *port*)

Set the host and port destination used for sending the SIP message. This can be useful for an application with 'DIRECT ROOTING MODE' NOTE: Instead, you should use the 'Route' header facility which leads to the same behaviour.

**Parameters:**

*ict* The element to work on.  
*destination* The destination host.  
*port* The destination port.

##### 4.4.4.2 int nict\_set\_destination (nict\_t \* *nict*, char \* *destination*, int *port*)

Set the host and port destination used for sending the SIP message. This can be useful for an application with 'DIRECT ROOTING MODE' NOTE: Instead, you should use the 'Route' header facility which leads to the same behaviour.

**Parameters:**

*nict* The element to work on.  
*destination* The destination host.  
*port* The destination port.

##### 4.4.4.3 sip\_event\_t\* nist\_need\_timer\_j\_event (nist\_t \* *nist*, state\_t *state*, int *transactionid*)

Check if this transaction needs a TIMEOUT\_J event

**Parameters:**

*nist* The element to work on.  
*state* The actual state of the transaction.  
*transactionid* The transaction id.

**4.4.4.4 transaction\_t\* osip\_create\_transaction (osip\_t \* *osip*, sipevent\_t \* *evt*)**

Create a transaction for this event (MUST be a SIP REQUEST event).

**Parameters:**

*osip* The element to work on.

*evt* The element representing the new SIP REQUEST.

**4.4.4.5 transaction\_t\* osip\_find\_transaction (osip\_t \* *osip*, sipevent\_t \* *evt*)**

Search for a transaction that match this event (MUST be a MESSAGE event).

**Parameters:**

*osip* The element to work on.

*evt* The element representing the SIP MESSAGE.

**4.4.4.6 void osip\_free (osip\_t \* *osip*)**

Free all resource in a osip\_t element.

**Parameters:**

*osip* The element to free.

**4.4.4.7 void osip\_global\_free ()**

Free all global resource hold by the oSIP stack. This can only be called after all osip\_t element has been "stopped".

**4.4.4.8 int osip\_global\_init ()**

Initialise the global oSIP stack elements. This method initialise the parser and load the fsm. This method MUST be called before any call to oSIP is made.

**4.4.4.9 int osip\_ict\_execute (osip\_t \* *osip*)**

Consume ALL pending sipevent\_t previously added in the fifos of ict transactions.

**Parameters:**

*osip* The element to work on.

**4.4.4.10 int osip\_init (osip\_t \*\* *osip*)**

Allocate an osip\_t element.

**Parameters:**

*osip* the element to allocate.

**4.4.4.11 int osip\_ist\_execute (osip\_t \* *osip*)**

Consume ALL pending sipevent\_t previously added in the fifos of ist transactions.

**Parameters:**

*osip* The element to work on.

**4.4.4.12 sipevent\_t\* osip\_new\_outgoing\_sipmessage (sip\_t \* *sip*)**

Allocate a sipevent (we know this message is an OUTGOING SIP message).

**Parameters:**

*sip* The SIP message we want to send.

**4.4.4.13 int osip\_nict\_execute (osip\_t \* *osip*)**

Consume ALL pending sipevent\_t previously added in the fifos of nict transactions.

**Parameters:**

*osip* The element to work on.

**4.4.4.14 int osip\_nist\_execute (osip\_t \* *osip*)**

Consume ALL pending sipevent\_t previously added in the fifos of nist transactions.

**Parameters:**

*osip* The element to work on.

**4.4.4.15 sipevent\_t\* osip\_parse (char \* *buf*)**

Create a sipevent from a SIP message string.

**Parameters:**

*buf* The SIP message as a string.

**4.4.4.16 void osip\_setcb\_ict\_1xx\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))**

Register the callback called when a 1xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.17** void osip\_setcb\_ict\_2xx\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 2xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.18** void osip\_setcb\_ict\_2xx\_received2 (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 2xx SIP message is received again. NOTE: obsolete... THIS IS NEVER CALLED! as the transaction is destroyed when the first 200 is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.19** void osip\_setcb\_ict\_3456xx\_received2 (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a retransmission of a final response is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.20** void osip\_setcb\_ict\_3xx\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 3xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.21** void osip\_setcb\_ict\_4xx\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 4xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.22** `void osip_setcb_ict_5xx_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 5xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.23** `void osip_setcb_ict_6xx_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 6xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.24** `void osip_setcb_ict_ack_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an ACK is sent. NOTE: This method is only called if the final response was not a 2xx

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.25** `void osip_setcb_ict_ack_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an ACK is retransmitted. NOTE: This method is only called if the final response was not a 2xx

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.26** `void osip_setcb_ict_invite_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an INVITE is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.



**4.4.4.27** `void osip_setcb_ict_invite_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an INVITE is retransmitted.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.28** `void osip_setcb_ict_kill_transaction (osip_t * cf, void(* cb)(transaction_t *))`

Register the callback called when the transaction is deleted.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.29** `void osip_setcb_ict_transport_error (osip_t * cf, void(* cb)(transaction_t *, int error))`

Register the callback called when a transport error happens.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.30** `void osip_setcb_list_1xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 1xx SIP message is sent.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.31** `void osip_setcb_list_1xx_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 1xx SIP message is sent again.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.32** `void osip_setcb_list_2xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is sent.

**Parameters:**

- cf* The osip element attached to the transaction.
- cb* The method we want to register.

**4.4.4.33** `void osip_setcb_list_2xx_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is sent again. NOTE: This method is never called because the transaction is destroyed right after the first 200 OK is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.34** `void osip_setcb_list_3456xx_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a final response (not 200) is sent again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.35** `void osip_setcb_list_3xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 3xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.36** `void osip_setcb_list_4xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 4xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.37** `void osip_setcb_list_5xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 5xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.38** `void osip_setcb_list_6xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 6xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.39** void osip\_setcb\_list\_ack\_received (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an ACK is received. NOTE: This method is only called if the final response was not a 2xx

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.40** void osip\_setcb\_list\_ack\_received2 (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an ACK is received again. NOTE: This method is only called if the final response was not a 2xx

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.41** void osip\_setcb\_list\_invite\_received (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an INVITE is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.42** void osip\_setcb\_list\_invite\_received2 (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an INVITE is received again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.43** void osip\_setcb\_list\_kill\_transaction (osip\_t \* cf, void(\* cb)(transaction\_t \*))

Register the callback called when the transaction is deleted.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.44** `void osip_setcb_ist_transport_error (osip_t * cf, void(* cb)(transaction_t *, int error))`

Register the callback called when a transport error happens.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.45** `void osip_setcb_nict_1xx_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 1xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.46** `void osip_setcb_nict_2xx_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.47** `void osip_setcb_nict_2xx_received2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is received again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.48** `void osip_setcb_nict_3456xx_received2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a final response (not 200) is received again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.49** void `osip_setcb_nict_3xx_received` (`osip_t * cf`, `void(* cb)(transaction_t *, sip_t *)`)

Register the callback called when a 3xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.50** void `osip_setcb_nict_4xx_received` (`osip_t * cf`, `void(* cb)(transaction_t *, sip_t *)`)

Register the callback called when a 4xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.51** void `osip_setcb_nict_5xx_received` (`osip_t * cf`, `void(* cb)(transaction_t *, sip_t *)`)

Register the callback called when a 5xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.52** void `osip_setcb_nict_6xx_received` (`osip_t * cf`, `void(* cb)(transaction_t *, sip_t *)`)

Register the callback called when a 6xx SIP message is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.53** void `osip_setcb_nict_bye_sent` (`osip_t * cf`, `void(* cb)(transaction_t *, sip_t *)`)

Register the callback called when an BYE is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.54** `void osip_setcb_nict_cancel_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an CANCEL is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.55** `void osip_setcb_nict_info_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an INFO is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.56** `void osip_setcb_nict_kill_transaction (osip_t * cf, void(* cb)(transaction_t *))`

Register the callback called when the transaction is deleted.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.57** `void osip_setcb_nict_notify_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an NOTIFY is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.58** `void osip_setcb_nict_options_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an OPTIONS is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.59** void osip\_setcb\_nict\_register\_sent (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an REGISTER is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.60** void osip\_setcb\_nict\_request\_sent2 (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an REQUEST is sent again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.61** void osip\_setcb\_nict\_subscribe\_sent (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an SUBSCRIBE is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.62** void osip\_setcb\_nict\_transport\_error (osip\_t \* cf, void(\* cb)(transaction\_t \*, int error))

Register the callback called when a transport error happens.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.63** void osip\_setcb\_nict\_unknown\_sent (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an UNKNOWN REQUEST is sent. NOTE: All SIP request that do not have specific callback will use this one.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.64** `void osip_setcb_nist_1xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 1xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.65** `void osip_setcb_nist_2xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.66** `void osip_setcb_nist_2xx_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 2xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.67** `void osip_setcb_nist_3456xx_sent2 (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a final response is sent again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.68** `void osip_setcb_nist_3xx_sent (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when a 3xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.



**4.4.4.69** void osip\_setcb\_nist\_4xx\_sent (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 4xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.70** void osip\_setcb\_nist\_5xx\_sent (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 5xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.71** void osip\_setcb\_nist\_6xx\_sent (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when a 6xx SIP message is sent.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.72** void osip\_setcb\_nist\_bye\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when an BYE is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.73** void osip\_setcb\_nist\_cancel\_received (osip\_t \* *cf*, void(\* *cb*)(transaction\_t \*, sip\_t \*))

Register the callback called when an CANCEL is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.74** `void osip_setcb_nist_info_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an INFO is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.75** `void osip_setcb_nist_kill_transaction (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when the transaction is deleted.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.76** `void osip_setcb_nist_notify_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an NOTIFY is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.77** `void osip_setcb_nist_options_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an OPTIONS is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.78** `void osip_setcb_nist_register_received (osip_t * cf, void(* cb)(transaction_t *, sip_t *))`

Register the callback called when an REGISTER is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.79** void osip\_setcb\_nist\_request\_received2 (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when a REQUEST is received again.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.80** void osip\_setcb\_nist\_subscribe\_received (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an SUBSCRIBE is received.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.81** void osip\_setcb\_nist\_transport\_error (osip\_t \* cf, void(\* cb)(transaction\_t \*, int error))

Register the callback called when a transport error happens.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.82** void osip\_setcb\_nist\_unknown\_received (osip\_t \* cf, void(\* cb)(transaction\_t \*, sip\_t \*))

Register the callback called when an unknown REQUEST is received. NOTE: When the message does not have a specific callback, this callback is used instead.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.83** void osip\_setcb\_send\_message (osip\_t \* cf, int(\* cb)(transaction\_t \*, sip\_t \*, char \*, int, int))

Register the callback used to send SIP message.

**Parameters:**

*cf* The osip element attached to the transaction.

*cb* The method we want to register.

**4.4.4.84 void osip\_timers\_ict\_execute (osip\_t \* *osip*)**

Check if an ict transactions needs a timer event.

**Parameters:**

*osip* The element to work on.

**4.4.4.85 void osip\_timers\_list\_execute (osip\_t \* *osip*)**

Check if an list transactions needs a timer event.

**Parameters:**

*osip* The element to work on.

**4.4.4.86 void osip\_timers\_nict\_execute (osip\_t \* *osip*)**

Check if a nict transactions needs a timer event.

**Parameters:**

*osip* The element to work on.

**4.4.4.87 void osip\_timers\_nist\_execute (osip\_t \* *osip*)**

Check if a nist transactions needs a timer event.

**Parameters:**

*osip* The element to work on.

**4.4.4.88 transaction\_t\* osip\_transaction\_find (list\_t \* *transactions*, sipevent\_t \* *evt*)**

Search for a transaction that match this event (MUST be a MESSAGE event).

**Parameters:**

*transactions* The list of transactions to work on.

*evt* The element representing the SIP MESSAGE.

**4.4.4.89 int transaction\_add\_event (transaction\_t \* *transaction*, sipevent\_t \* *evt*)**

Add a SIP event in the fifo of a transaction\_t element.

**Parameters:**

*transaction* The element to work on.

*evt* The event to add.

**4.4.4.90 int transaction\_execute (transaction\_t \* *transaction*, sip\_event\_t \* *evt*)**

Consume one sip\_event\_t element previously added in the fifo. NOTE: This method MUST NEVER be called within another call of this method. (For example, you can't call **transaction\_execute()** (p. 37) in a callback registered in the osip\_t element.)

**Parameters:**

*transaction* The element to free.

*evt* The element to consume.

**4.4.4.91 int transaction\_free (transaction\_t \* *transaction*)**

Free all resource in a transaction\_t element.

**Parameters:**

*transaction* The element to free.

**4.4.4.92 void\* transaction\_get\_your\_instance (transaction\_t \* *transaction*)**

Get a pointer to your personal context associated with this transaction.

**Parameters:**

*transaction* The element to work on.

**4.4.4.93 int transaction\_init (transaction\_t \*\* *transaction*, context\_type\_t *ctx\_type*, osip\_t \* *osip*, sip\_t \* *request*)**

Allocate an transaction\_t element.

**Parameters:**

*transaction* The element to allocate.

*ctx\_type* The type of transaction. (ICT, IST, NICT, NIST)

*osip* The global instance of oSIP.

*request* The SIP request that initiate the transaction.

**4.4.4.94 int transaction\_set\_your\_instance (transaction\_t \* *transaction*, void \* *instance*)**

Set a pointer to your personal context associated with this transaction. NOTE: this is a very useful method that allow you to avoid searching for your personal context inside the registered callbacks. You can initialise this pointer to your context right after the creation of the transaction\_t element. Then, you'll be able to get the address of your context by calling **transaction\_get\_your\_instance()** (p. 21).

**Parameters:**

*transaction* The element to work on.

*instance* The address of your context.

## 4.5 oSIP SDP parser Handling

### Compounds

- struct **sdp\_attribute\_t**
- struct **sdp\_bandwidth\_t**
- struct **sdp\_connection\_t**
- struct **sdp\_key\_t**
- struct **sdp\_media\_t**
- struct **sdp\_t**
- struct **sdp\_time\_descr\_t**

### Typedefs

- typedef sdp\_bandwidth\_t **sdp\_bandwidth\_t**
- typedef sdp\_time\_descr\_t **sdp\_time\_descr\_t**
- typedef sdp\_key\_t **sdp\_key\_t**
- typedef sdp\_attribute\_t **sdp\_attribute\_t**
- typedef sdp\_connection\_t **sdp\_connection\_t**
- typedef sdp\_media\_t **sdp\_media\_t**
- typedef sdp\_t **sdp\_t**

### Functions

- int **sdp\_bandwidth\_init** (sdp\_bandwidth\_t \*\*elem)
- void **sdp\_bandwidth\_free** (sdp\_bandwidth\_t \*elem)
- int **sdp\_time\_descr\_init** (sdp\_time\_descr\_t \*\*elem)
- void **sdp\_time\_descr\_free** (sdp\_time\_descr\_t \*elem)
- int **sdp\_key\_init** (sdp\_key\_t \*\*elem)
- void **sdp\_key\_free** (sdp\_key\_t \*elem)
- int **sdp\_attribute\_init** (sdp\_attribute\_t \*\*elem)
- void **sdp\_attribute\_free** (sdp\_attribute\_t \*elem)
- int **sdp\_connection\_init** (sdp\_connection\_t \*\*elem)
- void **sdp\_connection\_free** (sdp\_connection\_t \*elem)
- int **sdp\_media\_init** (sdp\_media\_t \*\*elem)
- void **sdp\_media\_free** (sdp\_media\_t \*elem)
- int **sdp\_init** (sdp\_t \*\*sdp)
- int **sdp\_parse** (sdp\_t \*sdp, const char \*buf)
- int **sdp\_2char** (sdp\_t \*sdp, char \*\*dest)
- void **sdp\_free** (sdp\_t \*sdp)
- int **sdp\_v\_version\_set** (sdp\_t \*sdp, char \*value)
- char \* **sdp\_v\_version\_get** (sdp\_t \*sdp)
- int **sdp\_o\_origin\_set** (sdp\_t \*sdp, char \*username, char \*sess\_id, char \*sess\_version, char \*nettype, char \*addrtype, char \*addr)
- char \* **sdp\_o\_username\_get** (sdp\_t \*sdp)
- char \* **sdp\_o\_sess\_id\_get** (sdp\_t \*sdp)
- char \* **sdp\_o\_sess\_version\_get** (sdp\_t \*sdp)
- char \* **sdp\_o\_nettype\_get** (sdp\_t \*sdp)
- char \* **sdp\_o\_addrtype\_get** (sdp\_t \*sdp)

- `char * sdp_o_addr_get (sdp_t *sdp)`
- `int sdp_s_name_set (sdp_t *sdp, char *value)`
- `char * sdp_s_name_get (sdp_t *sdp)`
- `int sdp_i_info_set (sdp_t *sdp, int pos_media, char *value)`
- `char * sdp_i_info_get (sdp_t *sdp, int pos_media)`
- `int sdp_u_uri_set (sdp_t *sdp, char *value)`
- `char * sdp_u_uri_get (sdp_t *sdp)`
- `int sdp_e_email_add (sdp_t *sdp, char *value)`
- `char * sdp_e_email_get (sdp_t *sdp, int pos)`
- `int sdp_p_phone_add (sdp_t *sdp, char *value)`
- `char * sdp_p_phone_get (sdp_t *sdp, int pos)`
- `int sdp_c_connection_add (sdp_t *sdp, int pos_media, char *nettype, char *addrtype, char *addr, char *addr_multicast_ttl, char *addr_multicast_int)`
- `char * sdp_c_nettype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addrtype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_multicast_ttl_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_multicast_int_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_b_bandwidth_add (sdp_t *sdp, int pos_media, char *bwtype, char *bandwidth)`
- `sdp_bandwidth_t * sdp_bandwidth_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_b_bwtype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_b_bandwidth_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_t_time_descr_add (sdp_t *sdp, char *start, char *stop)`
- `char * sdp_t_start_time_get (sdp_t *sdp, int pos_td)`
- `char * sdp_t_stop_time_get (sdp_t *sdp, int pos_td)`
- `int sdp_r_repeat_add (sdp_t *sdp, int pos_time_descr, char *value)`
- `char * sdp_r_repeat_get (sdp_t *sdp, int pos_time_descr, int pos_repeat)`
- `int sdp_z_adjustments_set (sdp_t *sdp, char *value)`
- `char * sdp_z_adjustments_get (sdp_t *sdp)`
- `int sdp_k_key_set (sdp_t *sdp, int pos_media, char *keytype, char *keydata)`
- `char * sdp_k_keytype_get (sdp_t *sdp, int pos_media)`
- `char * sdp_k_keydata_get (sdp_t *sdp, int pos_media)`
- `int sdp_a_attribute_add (sdp_t *sdp, int pos_media, char *att_field, char *att_value)`
- `sdp_attribute_t * sdp_attribute_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_a_att_field_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_a_att_value_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_endof_media (sdp_t *sdp, int pos)`
- `int sdp_m_media_add (sdp_t *sdp, char *media, char *port, char *number_of_port, char *proto)`
- `char * sdp_m_media_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_port_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_number_of_port_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_proto_get (sdp_t *sdp, int pos_media)`
- `int sdp_m_payload_add (sdp_t *sdp, int pos_media, char *payload)`
- `char * sdp_m_payload_get (sdp_t *sdp, int pos_media, int pos)`

## 4.5.1 Typedef Documentation

### 4.5.1.1 typedef struct sdp\_attribute\_t sdp\_attribute\_t

Structure for referencing an attribute header. @defvar sdp\_attribute\_t

#### 4.5.1.2 typedef struct sdp\_bandwidth\_t sdp\_bandwidth\_t

Structure for referencing bandwidth header. @defvar sdp\_bandwidth\_t

#### 4.5.1.3 typedef struct sdp\_connection\_t sdp\_connection\_t

Structure for referencing a connection header. @defvar sdp\_connection\_t

#### 4.5.1.4 typedef struct sdp\_key\_t sdp\_key\_t

Structure for referencing key header. @defvar sdp\_key\_t

#### 4.5.1.5 typedef struct sdp\_media\_t sdp\_media\_t

Structure for referencing a media header. @defvar sdp\_media\_t

#### 4.5.1.6 typedef struct sdp\_t sdp\_t

Structure for referencing a SDP packet. @defvar sdp\_t

#### 4.5.1.7 typedef struct sdp\_time\_descr\_t sdp\_time\_descr\_t

Structure for referencing time description header. @defvar sdp\_time\_descr\_t

### 4.5.2 Function Documentation

#### 4.5.2.1 int sdp\_2char (sdp\_t \* *sdp*, char \*\* *dest*)

Get a string representation of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*dest* The resulting new allocated buffer.

#### 4.5.2.2 char\* sdp\_a\_att\_field\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)

Get the attribute name ('a' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

*pos* The attribute line number.



**4.5.2.3 char\* sdp\_a\_att\_value\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get the attribute value ('a' field) of a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos\_media* The media line number.
- pos* The attribute line number.

**4.5.2.4 int sdp\_a\_attribute\_add (sdp\_t \* *sdp*, int *pos\_media*, char \* *att\_field*, char \* *att\_value*)**

Set the version in a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos\_media* The line number.
- att\_field* The token value.
- att\_value* The token value.

**4.5.2.5 void sdp\_attribute\_free (sdp\_attribute\_t \* *elem*)**

Free a attribute element.

**Parameters:**

- elem* The element to work on.

**4.5.2.6 sdp\_attribute\_t\* sdp\_attribute\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get one of the attribute ('a' field) of a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos\_media* The media line number.
- pos* The attribute line number.

**4.5.2.7 int sdp\_attribute\_init (sdp\_attribute\_t \*\* *elem*)**

Allocate an attribute element.

**Parameters:**

- elem* The element to work on.

#### 4.5.2.8 `int sdp_b_bandwidth_add (sdp_t * sdp, int pos_media, char * bwtype, char * bandwidth)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*bwtype* The token value.  
*bandwidth* The token value.

#### 4.5.2.9 `char* sdp_b_bandwidth_get (sdp_t * sdp, int pos_media, int pos)`

Get the bandwidth value ('b' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*pos* The index in the bandwidth element list..

#### 4.5.2.10 `char* sdp_b_bwtype_get (sdp_t * sdp, int pos_media, int pos)`

Get the bandwidth type ('b' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*pos* The index in the bandwidth element list..

#### 4.5.2.11 `void sdp_bandwidth_free (sdp_bandwidth_t * elem)`

Free a bandwidth element.

**Parameters:**

*elem* The element to work on.

#### 4.5.2.12 `sdp_bandwidth_t* sdp_bandwidth_get (sdp_t * sdp, int pos_media, int pos)`

Get the bandwidth ('b' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*pos* The index in the bandwidth element list..

**4.5.2.13 int sdp\_bandwidth\_init (sdp\_bandwidth\_t \*\* *elem*)**

Allocate a bandwidth element.

**Parameters:**

*elem* The element to work on.

**4.5.2.14 char\* sdp\_c\_addr\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get the address ('c' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

*pos* The index in the connection element list..

**4.5.2.15 char\* sdp\_c\_addr\_multicast\_int\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get the multicast int info ('c' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

*pos* The index in the connection element list..

**4.5.2.16 char\* sdp\_c\_addr\_multicast\_ttl\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get the multicast ttl ('c' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

*pos* The index in the connection element list..

**4.5.2.17 char\* sdp\_c\_addrtype\_get (sdp\_t \* *sdp*, int *pos\_media*, int *pos*)**

Get the address type ('c' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

*pos* The index in the connection element list..

**4.5.2.18** `int sdp_c_connection_add (sdp_t * sdp, int pos_media, char * nettype, char * addrtype, char * addr, char * addr_multicast_ttl, char * addr_multicast_int)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*nettype* The token value.  
*addrtype* The token value.  
*addr* The token value.  
*addr\_multicast\_ttl* The token value.  
*addr\_multicast\_int* The token value.

**4.5.2.19** `char* sdp_c_nettype_get (sdp_t * sdp, int pos_media, int pos)`

Get the network type ('c' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_media* The media line number.  
*pos* The index in the connection element list..

**4.5.2.20** `void sdp_connection_free (sdp_connection_t * elem)`

Free a connection element.

**Parameters:**

*elem* The element to work on.

**4.5.2.21** `int sdp_connection_init (sdp_connection_t ** elem)`

Allocate a connection element.

**Parameters:**

*elem* The element to work on.

**4.5.2.22** `int sdp_e_email_add (sdp_t * sdp, char * value)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*value* The token value.

**4.5.2.23** `char* sdp_e_email_get (sdp_t * sdp, int pos)`

Get one of the email ('e' field) of a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos* the index of the email line.

**4.5.2.24** `int sdp_endof_media (sdp_t * sdp, int pos)`

Check if there is more media lines a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos* The attribute line number.

**4.5.2.25** `void sdp_free (sdp_t * sdp)`

Free a SDP packet.

**Parameters:**

- sdp* The element to work on.

**4.5.2.26** `char* sdp_i_info_get (sdp_t * sdp, int pos_media)`

Get the session info ('i' field) of a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos\_media* The media line number.

**4.5.2.27** `int sdp_i_info_set (sdp_t * sdp, int pos_media, char * value)`

Set the version in a SDP packet.

**Parameters:**

- sdp* The element to work on.
- pos\_media* The media line number.
- value* The token value.

**4.5.2.28** `int sdp_init (sdp_t ** sdp)`

Allocate a SDP packet.

**Parameters:**

- sdp* The element to work on.

**4.5.2.29** `int sdp_k_key_set (sdp_t * sdp, int pos_media, char * keytype, char * keydata)`

Add a key in a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* index of m field.

*keytype* The token value.

*keydata* The token value.

**4.5.2.30** `char* sdp_k_keydata_get (sdp_t * sdp, int pos_media)`

Get the key value ('k' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

**4.5.2.31** `char* sdp_k_keytype_get (sdp_t * sdp, int pos_media)`

Get the key type ('k' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The media line number.

**4.5.2.32** `void sdp_key_free (sdp_key_t * elem)`

Free a key element.

**Parameters:**

*elem* The element to work on.

**4.5.2.33** `int sdp_key_init (sdp_key_t ** elem)`

Allocate a key element.

**Parameters:**

*elem* The element to work on.

**4.5.2.34** `int sdp_m_media_add (sdp_t * sdp, char * media, char * port, char * number_of_port, char * proto)`

Add a media line in a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***media*** The token value.  
***port*** The token value.  
***number\_of\_port*** The token value.  
***proto*** The token value.

**4.5.2.35 char\* sdp\_m\_media\_get (sdp\_t \* sdp, int pos\_media)**

Get the media type ('m' field) of a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***pos\_media*** The line number.

**4.5.2.36 char\* sdp\_m\_number\_of\_port\_get (sdp\_t \* sdp, int pos\_media)**

Get the number of port ('m' field) of a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***pos\_media*** The line number.

**4.5.2.37 int sdp\_m\_payload\_add (sdp\_t \* sdp, int pos\_media, char \* payload)**

Set the payload in a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***pos\_media*** The line number.  
***payload*** The token value.

**4.5.2.38 char\* sdp\_m\_payload\_get (sdp\_t \* sdp, int pos\_media, int pos)**

Get one of the payload number ('m' field) of a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***pos\_media*** The line number.  
***pos*** The i th payload element.

**4.5.2.39 char\* sdp\_m\_port\_get (sdp\_t \* sdp, int pos\_media)**

Get the port number ('m' field) of a SDP packet.

**Parameters:**

***sdp*** The element to work on.  
***pos\_media*** The line number.

**4.5.2.40** `char* sdp_m_proto_get (sdp_t * sdp, int pos_media)`

Get the protocol ('m' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_media* The line number.

**4.5.2.41** `void sdp_media_free (sdp_media_t * elem)`

Free a media element.

**Parameters:**

*elem* The element to work on.

**4.5.2.42** `int sdp_media_init (sdp_media_t ** elem)`

Allocate a media element.

**Parameters:**

*elem* The element to work on.

**4.5.2.43** `char* sdp_o_addr_get (sdp_t * sdp)`

Get the addr ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.44** `char* sdp_o_addrtype_get (sdp_t * sdp)`

Get the addrtype ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.45** `char* sdp_o_nettype_get (sdp_t * sdp)`

Get the nettype ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.



**4.5.2.46** `int sdp_o_origin_set (sdp_t * sdp, char * username, char * sess_id, char * sess_version, char * nettype, char * addrtype, char * addr)`

Set the origin field in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*username* The token value.  
*sess\_id* The token value.  
*sess\_version* The token value.  
*nettype* The token value.  
*addrtype* The token value.  
*addr* The token value.

**4.5.2.47** `char* sdp_o_sess_id_get (sdp_t * sdp)`

Get the session id ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.48** `char* sdp_o_sess_version_get (sdp_t * sdp)`

Get the session version ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.49** `char* sdp_o_username_get (sdp_t * sdp)`

Get the username ('o' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.50** `int sdp_p_phone_add (sdp_t * sdp, char * value)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*value* The token value.

**4.5.2.51 char\* sdp\_p\_phone\_get (sdp\_t \* *sdp*, int *pos*)**

Get one of the phone ('p' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos* the index of the phone line.

**4.5.2.52 int sdp\_parse (sdp\_t \* *sdp*, const char \* *buf*)**

Parse a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*buf* The buffer to parse.

**4.5.2.53 int sdp\_r\_repeat\_add (sdp\_t \* *sdp*, int *pos\_time\_descr*, char \* *value*)**

Set the repeat information ('r' field) in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_time\_descr* index of t field.  
*value* The token value.

**4.5.2.54 char\* sdp\_r\_repeat\_get (sdp\_t \* *sdp*, int *pos\_time\_descr*, int *pos\_repeat*)**

Get the repeat information ('r' field) in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*pos\_time\_descr* index of t field.  
*pos\_repeat* index of element in the 'r' field.

**4.5.2.55 char\* sdp\_s\_name\_get (sdp\_t \* *sdp*)**

Get the session name ('s' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.56 int sdp\_s\_name\_set (sdp\_t \* *sdp*, char \* *value*)**

Set the session name in a SDP packet.

**Parameters:**

*sdp* The element to work on.  
*value* The token value.

**4.5.2.57** `char* sdp_t_start_time_get (sdp_t * sdp, int pos_td)`

Get the start time value ('t' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_td* The time description line number.

**4.5.2.58** `char* sdp_t_stop_time_get (sdp_t * sdp, int pos_td)`

Get the stop time value ('t' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*pos\_td* The time description line number.

**4.5.2.59** `int sdp_t_time_descr_add (sdp_t * sdp, char * start, char * stop)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.

*start* The token value.

*stop* The token value.

**4.5.2.60** `void sdp_time_descr_free (sdp_time_descr_t * elem)`

Free a time description element.

**Parameters:**

*elem* The element to work on.

**4.5.2.61** `int sdp_time_descr_init (sdp_time_descr_t ** elem)`

Allocate a time description element.

**Parameters:**

*elem* The element to work on.

**4.5.2.62** `char* sdp_u_uri_get (sdp_t * sdp)`

Get the uri ('u' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.63** `int sdp_u_uri_set (sdp_t * sdp, char * value)`

Set the session info in a SDP packet.

**Parameters:**

*sdp* The element to work on.

*value* The token value.

**4.5.2.64** `char* sdp_v_version_get (sdp_t * sdp)`

Get the version ('v' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.65** `int sdp_v_version_set (sdp_t * sdp, char * value)`

Set the version in a SDP packet.

**Parameters:**

*sdp* The element to work on.

*value* The token value.

**4.5.2.66** `char* sdp_z_adjustments_get (sdp_t * sdp)`

Get the adjustments ('z' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

**4.5.2.67** `int sdp_z_adjustments_set (sdp_t * sdp, char * value)`

Set the adjustments ('z' field) of a SDP packet.

**Parameters:**

*sdp* The element to work on.

*value* The token value.

## 4.6 oSIP and SDP offer/answer model Handling

### Compounds

- struct **payload\_t**
- struct **sdp\_config\_t**
- struct **sdp\_context\_t**

### Typedefs

- typedef sdp\_context\_t **sdp\_context\_t**
- typedef payload\_t **payload\_t**
- typedef sdp\_config\_t **sdp\_config\_t**

### Functions

- int **sdp\_context\_init** (sdp\_context\_t \*\*ctx)
- void **sdp\_context\_free** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_mycontext** (sdp\_context\_t \*ctx, void \*value)
- void \* **sdp\_context\_get\_mycontext** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_local\_sdp** (sdp\_context\_t \*ctx, sdp\_t \*sdp)
- sdp\_t \* **sdp\_context\_get\_local\_sdp** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_remote\_sdp** (sdp\_context\_t \*ctx, sdp\_t \*sdp)
- sdp\_t \* **sdp\_context\_get\_remote\_sdp** (sdp\_context\_t \*ctx)
- int **payload\_init** (payload\_t \*\*payload)
- void **payload\_free** (payload\_t \*payload)
- int **sdp\_config\_init** ()
- void **sdp\_config\_free** ()
- int **sdp\_config\_set\_o\_username** (char \*tmp)
- int **sdp\_config\_set\_o\_session\_id** (char \*tmp)
- int **sdp\_config\_set\_o\_session\_version** (char \*tmp)
- int **sdp\_config\_set\_o\_nettype** (char \*tmp)
- int **sdp\_config\_set\_o\_addrtype** (char \*tmp)
- int **sdp\_config\_set\_o\_addr** (char \*tmp)
- int **sdp\_config\_set\_c\_nettype** (char \*tmp)
- int **sdp\_config\_set\_c\_addrtype** (char \*tmp)
- int **sdp\_config\_set\_c\_addr** (char \*tmp)
- int **sdp\_config\_set\_c\_addr\_multicast\_ttl** (char \*tmp)
- int **sdp\_config\_set\_c\_addr\_multicast\_int** (char \*tmp)
- int **sdp\_config\_add\_support\_for\_audio\_codec** (char \*payload, char \*number\_of\_port, char \*proto, char \*c\_nettype, char \*c\_addrtype, char \*c\_addr, char \*c\_addr\_multicast\_ttl, char \*c\_addr\_multicast\_int, char \*a\_rtpmap)
- int **sdp\_config\_add\_support\_for\_video\_codec** (char \*payload, char \*number\_of\_port, char \*proto, char \*c\_nettype, char \*c\_addrtype, char \*c\_addr, char \*c\_addr\_multicast\_ttl, char \*c\_addr\_multicast\_int, char \*a\_rtpmap)
- int **sdp\_config\_add\_support\_for\_other\_codec** (char \*payload, char \*number\_of\_port, char \*proto, char \*c\_nettype, char \*c\_addrtype, char \*c\_addr, char \*c\_addr\_multicast\_ttl, char \*c\_addr\_multicast\_int, char \*a\_rtpmap)
- int **sdp\_config\_set\_fcn\_set\_info** (int(\*fcn)(sdp\_context\_t \*, sdp\_t \*))
- int **sdp\_config\_set\_fcn\_set\_uri** (int(\*fcn)(sdp\_context\_t \*, sdp\_t \*))

- `int sdp_config_set_fcn_set_emails (int(*fcn)(sdp_context_t *, sdp_t *))`
- `int sdp_config_set_fcn_set_phones (int(*fcn)(sdp_context_t *, sdp_t *))`
- `int sdp_config_set_fcn_set_attributes (int(*fcn)(sdp_context_t *, sdp_t *, int))`
- `int sdp_config_set_fcn_accept_audio_codec (int(*fcn)(sdp_context_t *, char *, char *, int, char *))`
- `int sdp_config_set_fcn_accept_video_codec (int(*fcn)(sdp_context_t *, char *, char *, int, char *))`
- `int sdp_config_set_fcn_accept_other_codec (int(*fcn)(sdp_context_t *, char *, char *, char *, char *, char *))`
- `int sdp_config_set_fcn_get_audio_port (char *(*fcn)(sdp_context_t *, int))`
- `int sdp_config_set_fcn_get_video_port (char *(*fcn)(sdp_context_t *, int))`
- `int sdp_config_set_fcn_get_other_port (char *(*fcn)(sdp_context_t *, int))`
- `int sdp_context_execute_negotiation (sdp_context_t *ctx)`

## 4.6.1 Typedef Documentation

### 4.6.1.1 typedef struct payload\_t payload\_t

Structure for payload management. Each payload element represents one codec of a media line.  
@defvar payload\_t

### 4.6.1.2 typedef struct sdp\_config\_t sdp\_config\_t

Structure for storing the global configuration management. The information you store here is used when computing a remote SDP packet to build a compliant answer. The main objectives is to:  
\* automatically refuse unknown media. \* accept some of the known media. \* make sure the SDP answer match the SDP offer. \* simplify the SDP offer/answer model, as all unknown media are refused without any indication to the application layer. \* In any case, you can still modify the entire SDP packet after a negotiation if you are not satisfied by the negotiation result. @defvar sdp\_config\_t

### 4.6.1.3 typedef struct sdp\_context\_t sdp\_context\_t

Structure for applying the SDP offer/answer negotiation. The goal is simply to give: 1. A configuration (sdp\_config\_t) 2. A remote SDP packet (generally from the INVITE) The result is the creation of a local answer to the remote SDP packet. @defvar sdp\_context\_t

## 4.6.2 Function Documentation

### 4.6.2.1 void payload\_free (payload\_t \* *payload*)

Free a payload element.

#### Parameters:

*payload* The payload.

#### 4.6.2.2 int payload\_init (payload\_t \*\* *payload*)

Allocate a payload element.

**Parameters:**

*payload* The payload.

#### 4.6.2.3 int sdp\_config\_add\_support\_for\_audio\_codec (char \* *payload*, char \* *number\_of\_port*, char \* *proto*, char \* *c\_nettype*, char \* *c\_addrtype*, char \* *c\_addr*, char \* *c\_addr\_multicast\_ttl*, char \* *c\_addr\_multicast\_int*, char \* *a\_rtpmap*)

Add a supported audio codec. Those codecs will be accepted as long as you return 0 when the callback 'fcn\_accept\_audio\_codec' is called with the specific payload.

**Parameters:**

*payload* The payload.

*number\_of\_port* The number of port (channel) for this codec.

*proto* The protocol.

*c\_nettype* The network type in the 'c' field.

*c\_addrtype* The address type in the 'c' field.

*c\_addr* The address in the 'c' field.

*c\_addr\_multicast\_ttl* The ttl for multicast address in the 'c' field.

*c\_addr\_multicast\_int* The int for multicast address in the 'c' field.

*a\_rtpmap* The rtpmap attribute in the 'a' field.

#### 4.6.2.4 int sdp\_config\_add\_support\_for\_other\_codec (char \* *payload*, char \* *number\_of\_port*, char \* *proto*, char \* *c\_nettype*, char \* *c\_addrtype*, char \* *c\_addr*, char \* *c\_addr\_multicast\_ttl*, char \* *c\_addr\_multicast\_int*, char \* *a\_rtpmap*)

Add a supported (non-audio and non-video) codec. Those codecs will be accepted as long as you return 0 when the callback 'fcn\_accept\_other\_codec' is called with the specific payload.

**Parameters:**

*payload* The payload.

*number\_of\_port* The number of port (channel) for this codec.

*proto* The protocol.

*c\_nettype* The network type in the 'c' field.

*c\_addrtype* The address type in the 'c' field.

*c\_addr* The address in the 'c' field.

*c\_addr\_multicast\_ttl* The ttl for multicast address in the 'c' field.

*c\_addr\_multicast\_int* The int for multicast address in the 'c' field.

*a\_rtpmap* The rtpmap attribute in the 'a' field.

**4.6.2.5** `int sdp_config_add_support_for_video_codec (char * payload, char * number_of_port, char * proto, char * c_nettype, char * c_addrtype, char * c_addr, char * c_addr_multicast_ttl, char * c_addr_multicast_int, char * a_rtpmap)`

Add a supported video codec. Those codecs will be accepted as long as you return 0 when the callback 'fcn\_accept\_video\_codec' is called with the specific payload.

**Parameters:**

*payload* The payload.  
*number\_of\_port* The number of port (channel) for this codec.  
*proto* The protocol.  
*c\_nettype* The network type in the 'c' field.  
*c\_addrtype* The address type in the 'c' field.  
*c\_addr* The address in the 'c' field.  
*c\_addr\_multicast\_ttl* The ttl for multicast address in the 'c' field.  
*c\_addr\_multicast\_int* The int for multicast address in the 'c' field.  
*a\_rtpmap* The rtpmap attribute in the 'a' field.

**4.6.2.6** `void sdp_config_free ()`

Free resource stored by a sdp\_config element. This method must be called once when the application is stopped.

**4.6.2.7** `int sdp_config_init ()`

Initialise (and Allocate) a sdp\_config element (this element is global). This method must be called when the application is started.

**4.6.2.8** `int sdp_config_set_c_addr (char * tmp)`

Set the local IP address ('c' field) of all local SDP packet.

**Parameters:**

*tmp* The IP address.

**4.6.2.9** `int sdp_config_set_c_addr_multicast_int (char * tmp)`

Set the local int for multicast address ('c' field) of all local SDP packet.

**Parameters:**

*tmp* The int for multicast address.

**4.6.2.10** `int sdp_config_set_c_addr_multicast_ttl (char * tmp)`

Set the local ttl for multicast address ('c' field) of all local SDP packet.

**Parameters:**

*tmp* The ttl for multicast address.



**4.6.2.11 int sdp\_config\_set\_c\_addrtype (char \* *tmp*)**

Set the local address type ('c' field) of all local SDP packet.

**Parameters:**

*tmp* The address type.

**4.6.2.12 int sdp\_config\_set\_c\_nettype (char \* *tmp*)**

Set the local network type ('c' field) of all local SDP packet.

**Parameters:**

*tmp* The network type.

**4.6.2.13 int sdp\_config\_set\_fcn\_accept\_audio\_codec (int(\* *fcn*)(sdp\_context\_t \*, char \*, char \*, int, char \*))**

Set the callback used to accept a codec during a negotiation. This callback is called once each time we need to accept a codec.

**Parameters:**

*fcn* The callback.

**4.6.2.14 int sdp\_config\_set\_fcn\_accept\_other\_codec (int(\* *fcn*)(sdp\_context\_t \*, char \*, char \*, char \*, char \*))**

Set the callback used to accept a codec during a negotiation. This callback is called once each time we need to accept a codec.

**Parameters:**

*fcn* The callback.

**4.6.2.15 int sdp\_config\_set\_fcn\_accept\_video\_codec (int(\* *fcn*)(sdp\_context\_t \*, char \*, char \*, int, char \*))**

Set the callback used to accept a codec during a negotiation. This callback is called once each time we need to accept a codec.

**Parameters:**

*fcn* The callback.

**4.6.2.16 int sdp\_config\_set\_fcn\_get\_audio\_port (char \*(\* *fcn*)(sdp\_context\_t \*, int))**

Set the callback for setting the port number ('m' field) in a local SDP packet. This callback is called once each time a 'm' line is accepted.

**Parameters:**

*fcn* The callback.

**4.6.2.17 int sdp\_config\_set\_fcn\_get\_other\_port (char \*(\* fcn)(sdp\_context\_t \*, int))**

Set the callback for setting the port number ('m' field) in a local SDP packet. This callback is called once each time a 'm' line is accepted.

**Parameters:**

*fcn* The callback.

**4.6.2.18 int sdp\_config\_set\_fcn\_get\_video\_port (char \*(\* fcn)(sdp\_context\_t \*, int))**

Set the callback for setting the port number ('m' field) in a local SDP packet. This callback is called once each time a 'm' line is accepted.

**Parameters:**

*fcn* The callback.

**4.6.2.19 int sdp\_config\_set\_fcn\_set\_attributes (int(\* fcn)(sdp\_context\_t \*, sdp\_t \*, int))**

Set the callback for setting an attribute ('a' field) in a local SDP packet. This callback is called once each time we need an 'a' field.

**Parameters:**

*fcn* The callback.

**4.6.2.20 int sdp\_config\_set\_fcn\_set\_emails (int(\* fcn)(sdp\_context\_t \*, sdp\_t \*))**

Set the callback for setting an email ('e' field) in a local SDP packet. This callback is called once each time we need an 'e' field.

**Parameters:**

*fcn* The callback.

**4.6.2.21 int sdp\_config\_set\_fcn\_set\_info (int(\* fcn)(sdp\_context\_t \*, sdp\_t \*))**

Set the callback for setting info ('i' field) in a local SDP packet. This callback is called once each time we need an 'i' field.

**Parameters:**

*fcn* The callback.

**4.6.2.22 int sdp\_config\_set\_fcn\_set\_phones (int(\* fcn)(sdp\_context\_t \*, sdp\_t \*))**

Set the callback for setting a phone ('p' field) in a local SDP packet. This callback is called once each time we need an 'p' field.

**Parameters:**

*fcn* The callback.

**4.6.2.23 int sdp\_config\_set\_fcn\_set\_uri (int(\* fcn)(sdp\_context\_t \*, sdp\_t \*))**

Set the callback for setting a URI ('u' field) in a local SDP packet. This callback is called once each time we need an 'u' field.

**Parameters:**

*fcn* The callback.

**4.6.2.24 int sdp\_config\_set\_o\_addr (char \* tmp)**

Set the local IP address ('o' field) of all local SDP packet.

**Parameters:**

*tmp* The IP address.

**4.6.2.25 int sdp\_config\_set\_o\_addrtype (char \* tmp)**

Set the local address type ('o' field) of all local SDP packet.

**Parameters:**

*tmp* The address type.

**4.6.2.26 int sdp\_config\_set\_o\_nettype (char \* tmp)**

Set the local network type ('o' field) of all local SDP packet.

**Parameters:**

*tmp* The network type.

**4.6.2.27 int sdp\_config\_set\_o\_session\_id (char \* tmp)**

Set the local session id ('o' field) of all local SDP packet. WARNING: this field should be updated for each new SDP packet?

**Parameters:**

*tmp* The session id.

**4.6.2.28 int sdp\_config\_set\_o\_session\_version (char \* tmp)**

Set the local session version ('o' field) of all local SDP packet. WARNING: this field should be updated for each new SDP packet?

**Parameters:**

*tmp* The session version.

**4.6.2.29 int sdp\_config\_set\_o\_username (char \* *tmp*)**

Set the local username ('o' field) of all local SDP packet.

**Parameters:**

*tmp* The username.

**4.6.2.30 int sdp\_context\_execute\_negotiation (sdp\_context\_t \* *ctx*)**

Start the automatic negotiation for a UA NOTE: You can previously set context->mycontext to point to your personal context. This way you'll get access to your personal context in the callback and you can easily take the correct decisions. After this method is called, the negotiation will happen and callbacks will be called. You can modify, add, remove SDP fields, and accept and refuse the codec from your preferred list by using those callbacks. Of course, after the negotiation happen, you can modify the SDP packet if you wish to improve it or just refine some attributes.

**Parameters:**

*ctx* The context holding the remote SDP offer.

**4.6.2.31 void sdp\_context\_free (sdp\_context\_t \* *ctx*)**

Free a bandwidth element.

**Parameters:**

*ctx* The element to work on.

**4.6.2.32 sdp\_t\* sdp\_context\_get\_local\_sdp (sdp\_context\_t \* *ctx*)**

Get the local SDP packet associated to this negotiation.

**Parameters:**

*ctx* The element to work on.

**4.6.2.33 void\* sdp\_context\_get\_mycontext (sdp\_context\_t \* *ctx*)**

Get the context associated to this negotiation.

**Parameters:**

*ctx* The element to work on.

**4.6.2.34 sdp\_t\* sdp\_context\_get\_remote\_sdp (sdp\_context\_t \* *ctx*)**

Get the remote SDP packet associated to this negotiation.

**Parameters:**

*ctx* The element to work on.

**4.6.2.35 int sdp\_context\_init (sdp\_context\_t \*\* *ctx*)**

Allocate a bandwidth element.

**Parameters:**

*ctx* The element to work on.

**4.6.2.36 int sdp\_context\_set\_local\_sdp (sdp\_context\_t \* *ctx*, sdp\_t \* *sdp*)**

Set the local SDP packet associated to this negotiation. NOTE: This is done by the 'negotiator'. (You only need to give the remote SDP packet)

**Parameters:**

*ctx* The element to work on.

*sdp* The local SDP packet.

**4.6.2.37 int sdp\_context\_set\_mycontext (sdp\_context\_t \* *ctx*, void \* *value*)**

Set the context associated to this negotiation.

**Parameters:**

*ctx* The element to work on.

*value* A pointer to your personal context.

**4.6.2.38 int sdp\_context\_set\_remote\_sdp (sdp\_context\_t \* *ctx*, sdp\_t \* *sdp*)**

Set the remote SDP packet associated to this negotiation.

**Parameters:**

*ctx* The element to work on.

*sdp* The remote SDP packet.

## 4.7 oSIP semaphore definitions

### Typedefs

- typedef pthread\_mutex\_t **smutex\_t**
- typedef sem\_t **ssem\_t**

### Functions

- **smutex\_t \* smutex\_init** ()
- void **smutex\_destroy** (smutex\_t \*mut)
- int **smutex\_lock** (smutex\_t \*mut)
- int **smutex\_unlock** (smutex\_t \*mut)
- **ssem\_t \* ssem\_init** (unsigned int value)
- int **ssem\_destroy** (ssem\_t \*sem)
- int **ssem\_post** (ssem\_t \*sem)
- int **ssem\_wait** (ssem\_t \*sem)
- int **ssem\_trywait** (ssem\_t \*sem)

### 4.7.1 Typedef Documentation

#### 4.7.1.1 typedef pthread\_mutex\_t smutex\_t

Structure for referencing a semaphore element. @defvar smutex\_t

#### 4.7.1.2 typedef sem\_t ssem\_t

Structure for referencing a semaphore element. @defvar ssem\_t

### 4.7.2 Function Documentation

#### 4.7.2.1 void smutex\_destroy (smutex\_t \* *mut*)

Destroy the mutex.

**Parameters:**

*mut* The mutex to destroy.

#### 4.7.2.2 smutex\_t\* smutex\_init ()

Allocate and Initialise a semaphore.

#### 4.7.2.3 int smutex\_lock (smutex\_t \* *mut*)

Lock the mutex.

**Parameters:**

*mut* The mutex to lock.

**4.7.2.4 int smutex\_unlock (smutex\_t \* *mut*)**

Unlock the mutex.

**Parameters:**

*mut* The mutex to unlock.

**4.7.2.5 int ssem\_destroy (ssem\_t \* *sem*)**

Destroy a semaphore.

**Parameters:**

*sem* The semaphore to destroy.

**4.7.2.6 ssem\_t\* ssem\_init (unsigned int *value*)**

Allocate and Initialise a semaphore.

**Parameters:**

*value* The initial value for the semaphore.

**4.7.2.7 int ssem\_post (ssem\_t \* *sem*)**

Post operation on a semaphore.

**Parameters:**

*sem* The semaphore to destroy.

**4.7.2.8 int ssem\_trywait (ssem\_t \* *sem*)**

Wait operation on a semaphore. NOTE: if the semaphore is at 0, this call won't block.

**Parameters:**

*sem* The semaphore to destroy.

**4.7.2.9 int ssem\_wait (ssem\_t \* *sem*)**

Wait operation on a semaphore. NOTE: this call will block if the semaphore is at 0.

**Parameters:**

*sem* The semaphore to destroy.

## 4.8 oSIP parser Handling

### Defines

- `#define MSG_IS_RESPONSE(msg) ((msg) → strtline → statuscode!=NULL)`
- `#define MSG_IS_REQUEST(msg) ((msg) → strtline → statuscode==NULL)`
- `#define MSG_IS_INVITE(msg)`
- `#define MSG_IS_ACK(msg)`
- `#define MSG_IS_REGISTER(msg)`
- `#define MSG_IS_BYE(msg)`
- `#define MSG_IS_OPTIONS(msg)`
- `#define MSG_IS_INFO(msg)`
- `#define MSG_IS_CANCEL(msg)`
- `#define MSG_IS_NOTIFY(msg)`
- `#define MSG_IS_SUBSCRIBE(msg)`
- `#define MSG_IS_PRACK(msg)`
- `#define MSG_IS_STATUS_1XX(msg)`
- `#define MSG_IS_STATUS_2XX(msg)`
- `#define MSG_IS_STATUS_3XX(msg)`
- `#define MSG_IS_STATUS_4XX(msg)`
- `#define MSG_IS_STATUS_5XX(msg)`
- `#define MSG_IS_STATUS_6XX(msg)`
- `#define MSG_TEST_CODE(msg, code)`
- `#define MSG_IS_RESPONSEFOR(msg, requestname)`
- `#define generic_param_init(GP) url_param_init(GP)`
- `#define generic_param_free(GP) url_param_free(GP)`
- `#define generic_param_set(GP, NAME, VALUE) url_param_set(GP, NAME, VALUE)`
- `#define generic_param_clone(GP, DEST) url_param_clone(GP,DEST)`
- `#define generic_param_add(LIST, NAME, VALUE) url_param-add(LIST,NAME,VALUE)`
- `#define generic_param_getbyname(LIST, NAME, DEST) url_param-getbyname(LIST,NAME,DEST)`
- `#define accept_init(header) content_type_init(header)`
- `#define accept_free(header) content_type_free(header)`
- `#define accept_parse(header, hvalue) content_type_parse(header, hvalue)`
- `#define accept_2char(header, dest) content_type_2char(header, dest)`
- `#define accept_clone(header, dest) content_type_clone(header, dest)`
- `#define accept_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos, dest)`
- `#define accept_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define accept_param_getbyname(header, name, dest) generic_param-getbyname((header) → gen_params,name,dest)`
- `#define accept_encoding_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos, dest)`
- `#define accept_encoding_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define accept_encoding_param_getbyname(header, name, dest) generic_param-getbyname((header) → gen_params,name,dest)`
- `#define accept_language_init(header) accept_encoding_init(header)`
- `#define accept_language_parse(header, hvalue) accept_encoding_parse(header, hvalue)`



- `#define accept_language_2char(header, dest) accept_encoding_2char(header, dest)`
- `#define accept_language_free(header) accept_encoding_free(header)`
- `#define accept_language_clone(header, dest) accept_encoding_clone(header, dest)`
- `#define accept_language_getelement(header) accept_encoding_getelement(header)`
- `#define accept_language_setelement(header, value) accept_encoding_setelement(header, value)`
- `#define accept_language_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos, dest)`
- `#define accept_language_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define accept_language_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`
- `#define alert_info_init(header) call_info_init(header)`
- `#define alert_info_free(header) call_info_free(header)`
- `#define alert_info_parse(header, hvalue) call_info_parse(header, hvalue)`
- `#define alert_info_2char(header, dest) call_info_2char(header, dest)`
- `#define alert_info_clone(header, dest) call_info_clone(header, dest)`
- `#define alert_info_geturi(header) call_info_geturi(header)`
- `#define alert_info_seturi(header, uri) call_info_seturi(header, uri)`
- `#define allow_init(header) content_length_init(header)`
- `#define allow_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define allow_2char(header, dest) content_length_2char(header, dest)`
- `#define allow_free(header) content_length_free(header)`
- `#define allow_clone(header, dest) content_length_clone(header, dest)`
- `#define contact_getdisplayname(header) from_getdisplayname((from_t*)header)`
- `#define contact_setdisplayname(header, value) from_setdisplayname((from_t*)header, value)`
- `#define contact_geturl(header) from_geturl((from_t*)header)`
- `#define contact_seturl(header, url) from_seturl((from_t*)header, url)`
- `#define contact_param_get(header, pos, dest) from_param_get((from_t*)header, pos, dest)`
- `#define contact_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define contact_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`
- `#define content_disposition_init(header) call_info_init(header)`
- `#define content_disposition_free(header) call_info_free(header)`
- `#define content_disposition_2char(header, dest) call_info_2char(header, dest)`
- `#define content_disposition_clone(header, dest) call_info_clone(header, dest)`
- `#define content_disposition_settype(header, value) call_info_seturi(header, value)`
- `#define content_disposition_gettype(header) call_info_geturi(header)`
- `#define content_encoding_init(header) content_length_init(header)`
- `#define content_encoding_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define content_encoding_2char(header, dest) content_length_2char(header, dest)`
- `#define content_encoding_free(header) content_length_free(header)`
- `#define content_encoding_clone(header, dest) content_length_clone(header, dest)`
- `#define content_type_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos, dest)`
- `#define content_type_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define content_type_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`

- `#define error_info_init(header) call_info_init(header)`
- `#define error_info_free(header) call_info_free(header)`
- `#define error_info_parse(header, hvalue) call_info_parse(header, hvalue)`
- `#define error_info_2char(header, dest) call_info_2char(header, dest)`
- `#define error_info_clone(header, dest) call_info_clone(header, dest)`
- `#define error_info_seturi(header, uri) call_info_seturi(header, uri)`
- `#define error_info_geturi(header) call_info_geturi(header)`
- `#define from_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define from_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`
- `#define from_get_tag(header, dest) generic_param_getbyname((header) → gen_params, "tag", dest)`
- `#define from_set_tag(header, value) generic_param_add((header) → gen_params, sgetcopy("tag"), value)`
- `#define mime_version_init(header) content_length_init(header)`
- `#define mime_version_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define mime_version_2char(header, dest) content_length_2char(header, dest)`
- `#define mime_version_free(header) content_length_free(header)`
- `#define mime_version_clone(header, dest) content_length_clone(header, dest)`
- `#define proxy_authenticate_init(header) www_authenticate_init(header)`
- `#define proxy_authenticate_parse(header, hvalue) www_authenticate_parse(header, hvalue)`
- `#define proxy_authenticate_2char(header, dest) www_authenticate_2char(header, dest)`
- `#define proxy_authenticate_free(header) www_authenticate_free(header)`
- `#define proxy_authenticate_clone(header, dest) www_authenticate_clone(header, dest)`
- `#define proxy_authenticate_getauth_type(header) www_authenticate_getauth_type(header)`
- `#define proxy_authenticate_setauth_type(header, value) www_authenticate_setauth_type(header, value)`
- `#define proxy_authenticate_getrealm(header) www_authenticate_getrealm(header)`
- `#define proxy_authenticate_setrealm(header, value) www_authenticate_setrealm(header, value)`
- `#define proxy_authenticate_getdomain(header) www_authenticate_getdomain(header)`
- `#define proxy_authenticate_setdomain(header, value) www_authenticate_setdomain(header, value)`
- `#define proxy_authenticate_getnonce(header) www_authenticate_getnonce(header)`
- `#define proxy_authenticate_setnonce(header, value) www_authenticate_setnonce(header, value)`
- `#define proxy_authenticate_getopaque(header) www_authenticate_getopaque(header)`
- `#define proxy_authenticate_setopaque(header, value) www_authenticate_setopaque(header, value)`
- `#define proxy_authenticate_getstale(header) www_authenticate_getstale(header)`
- `#define proxy_authenticate_setstale(header, value) www_authenticate_setstale(header, value)`
- `#define proxy_authenticate_setstale_true(header) www_authenticate_setstale(header, sgetcopy("true"))`
- `#define proxy_authenticate_setstale_false(header) www_authenticate_setstale(header, sgetcopy("false"))`
- `#define proxy_authenticate_getalgorithm(header) www_authenticate_getalgorithm(header)`

- `#define proxy_authenticate_setalgorithm(header, value) www_authenticate_setalgorithm(header, value)`
- `#define proxy_authenticate_setalgorithm_MD5(header) www_authenticate_setalgorithm(header, sgetcopy("MD5"))`
- `#define proxy_authenticate_getqop_options(header) www_authenticate_getqop_options(header)`
- `#define proxy_authenticate_setqop_options(header, value) www_authenticate_setqop_options(header, value)`
- `#define proxy_authorization_init(header) authorization_init(header)`
- `#define proxy_authorization_parse(header, hvalue) authorization_parse(header, hvalue)`
- `#define proxy_authorization_2char(header, dest) authorization_2char(header, dest)`
- `#define proxy_authorization_free(header) authorization_free(header)`
- `#define proxy_authorization_clone(header, dest) authorization_clone(header, dest)`
- `#define proxy_authorization_getauth_type(header) authorization_getauth_type(header)`
- `#define proxy_authorization_setauth_type(header, value) authorization_setauth_type(header, value)`
- `#define proxy_authorization_getusername(header) authorization_getusername(header)`
- `#define proxy_authorization_setusername(header, value) authorization_setusername(header, value)`
- `#define proxy_authorization_getrealm(header) authorization_getrealm(header)`
- `#define proxy_authorization_setrealm(header, value) authorization_setrealm(header, value)`
- `#define proxy_authorization_getnonce(header) authorization_getnonce(header)`
- `#define proxy_authorization_setnonce(header, value) authorization_setnonce(header, value)`
- `#define proxy_authorization_geturi(header) authorization_geturi(header)`
- `#define proxy_authorization_seturi(header, value) authorization_seturi(header, value)`
- `#define proxy_authorization_getresponse(header) authorization_getresponse(header)`
- `#define proxy_authorization_setresponse(header, value) authorization_setresponse(header, value)`
- `#define proxy_authorization_getdigest(header) authorization_getdigest(header)`
- `#define proxy_authorization_setdigest(header, value) authorization_setdigest(header, value)`
- `#define proxy_authorization_getalgorithm(header) authorization_getalgorithm(header)`
- `#define proxy_authorization_setalgorithm(header, value) authorization_setalgorithm(header, value)`
- `#define proxy_authorization_getcnonce(header) authorization_getcnonce(header)`
- `#define proxy_authorization_setcnonce(header, value) authorization_setcnonce(header, value)`
- `#define proxy_authorization_getopaque(header) authorization_getopaque(header)`
- `#define proxy_authorization_setopaque(header, value) authorization_setopaque(header, value)`
- `#define proxy_authorization_getmessage_qop(header) authorization_getmessage_qop(header)`
- `#define proxy_authorization_setmessage_qop(header, value) authorization_setmessage_qop(header, value)`
- `#define proxy_authorization_getnonce_count(header) authorization_getnonce_count(header)`
- `#define proxy_authorization_setnonce_count(header, value) authorization_setnonce_count(header, value)`
- `#define record_route_clone(header, dest) from_clone(header, dest)`

- `#define record_route_seturl(header, url) from_seturl((from_t*)header,url)`
- `#define record_route_geturl(header) from_geturl((from_t*)header)`
- `#define record_route_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`
- `#define record_route_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define record_route_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define route_clone(header, dest) from_clone(header,dest)`
- `#define route_seturl(header, url) from_seturl((from_t*)header,url)`
- `#define route_geturl(header) from_geturl((from_t*)header)`
- `#define route_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`
- `#define route_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define route_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define to_setdisplayname(header, value) from_setdisplayname((from_t*)header,value)`
- `#define to_getdisplayname(header) from_getdisplayname((from_t*)header)`
- `#define to_seturl(header, url) from_seturl((from_t*)header,url)`
- `#define to_geturl(header) from_geturl((from_t*)header)`
- `#define to_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`
- `#define to_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define to_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define to_set_tag(header, value) generic_param_add((header) → gen_params, sgetcopy("tag"),value)`
- `#define to_get_tag(header, dest) generic_param_getbyname((header) → gen_params, "tag",dest)`
- `#define via_set_hidden(header) generic_param_add((header) → via_params,sgetcopy("hidden"),NULL)`
- `#define via_set_ttl(header, value) generic_param_add((header) → via_params,sgetcopy("ttl"),value)`
- `#define via_set_maddr(header, value) generic_param_add((header) → via_params,sgetcopy("maddr"),value)`
- `#define via_set_received(header, value) generic_param_add((header) → via_params,sgetcopy("received"),value)`
- `#define via_set_branch(header, value) generic_param_add((header) → via_params,sgetcopy("branch"),value)`
- `#define via_param_get(header, pos, dest) generic_param_get(header,pos,dest)`
- `#define via_param_add(header, name, value) generic_param_add((header) → via_params,name,value)`
- `#define via_param_getbyname(header, name, dest) generic_param_getbyname((header) → via_params,name,dest)`
- `#define www_authenticate_setstale_true(header) www_authenticate_setstale(header,sgetcopy("true"))`
- `#define www_authenticate_setstale_false(header) www_authenticate_setstale(header,sgetcopy("false"))`
- `#define www_authenticate_setalgorithm_MD5(header) www_authenticate_setalgorithm(header,sgetcopy("MD5"))`
- `#define msg_setdate(header, value) msg_setheader((sip_t *)header,(char *)"date",value)`

- `#define msg_getdate(header, pos, dest) msg_header_getbyname(( sip_t *)header, "date", pos, (header_t **)dest)`
- `#define msg_setencryption(header, value) msg_setheader((sip_t *)header, (char *)"encryption", value)`
- `#define msg_getencryption(header, pos, dest) msg_header_getbyname(( sip_t *)header, "encryption", pos, (header_t **)dest)`
- `#define msg_setorganization(header, value) msg_setheader((sip_t *)header, (char *)"organization", value)`
- `#define msg_getorganization(header, pos, dest) msg_header_getbyname(( sip_t *)header, "organization", pos, (header_t **)dest)`
- `#define msg_setrequire(header, value) msg_setheader((sip_t *)header, (char *)"require", value)`
- `#define msg_getrequire(header, pos, dest) msg_header_getbyname(( sip_t *)header, "require", pos, (header_t **)dest)`
- `#define msg_setsupported(header, value) msg_setheader((sip_t *)header, (char *)"supported", value)`
- `#define msg_getsupported(header, pos, dest) msg_header_getbyname(( sip_t *)header, "supported", pos, (header_t **)dest)`
- `#define msg_settimestamp(header, value) msg_setheader((sip_t *)header, (char *)"timestamp", value)`
- `#define msg_gettimestamp(header, pos, dest) msg_header_getbyname(( sip_t *)header, "timestamp", pos, (header_t **)dest)`
- `#define msg_setuser_agent(header, value) msg_setheader((sip_t *)header, (char *)"user-agent", value)`
- `#define msg_getuser_agent(header, pos, dest) msg_header_getbyname(( sip_t *)header, "user-agent", pos, (header_t **)dest)`
- `#define msg_setcontent_language(header, value) msg_setheader((sip_t *)header, (char *)"content-language", value)`
- `#define msg_getcontent_language(header, pos, dest) msg_header_getbyname(( sip_t *)header, "content-language", pos, (header_t **)dest)`
- `#define msg_setexpires(header, value) msg_setheader((sip_t *)header, (char *)"expires", value)`
- `#define msg_getexpires(header, pos, dest) msg_header_getbyname(( sip_t *)header, "expires", pos, (header_t **)dest)`
- `#define msg_setin_reply_to(header, value) msg_setheader((sip_t *)header, (char *)"in-reply-to", value)`
- `#define msg_getin_reply_to(header, pos, dest) msg_header_getbyname(( sip_t *)header, "in-reply-to", pos, (header_t **)dest)`
- `#define msg_setmax_forward(header, value) msg_setheader((sip_t *)header, (char *)"max-forward", value)`
- `#define msg_getmax_forward(header, pos, dest) msg_header_getbyname(( sip_t *)header, "max-forward", pos, (header_t **)dest)`
- `#define msg_setpriority(header, value) msg_setheader((sip_t *)header, (char *)"priority", value)`
- `#define msg_getpriority(header, pos, dest) msg_header_getbyname(( sip_t *)header, "priority", pos, (header_t **)dest)`
- `#define msg_setproxy_require(header, value) msg_setheader((sip_t *)header, (char *)"proxy-require", value)`
- `#define msg_getproxy_require(header, pos, dest) msg_header_getbyname(( sip_t *)header, "proxy-require", pos, (header_t **)dest)`
- `#define msg_setresponse_key(header, value) msg_setheader((sip_t *)header, (char *)"response-key", value)`

- #define **msg\_getresponse\_key**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"response-key",pos,(header\_t \*\*)dest)
- #define **msg\_setsubject**(header, value) msg\_setheader((sip\_t \*)header,(char \*)"subject",value)
- #define **msg\_getsubject**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"subject",pos,(header\_t \*\*)dest)
- #define **msg\_setretry\_after**(header, value) msg\_setheader((sip\_t \*)header,(char \*)"retry-after",value)
- #define **msg\_getretry\_after**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"retry-after",pos,(header\_t \*\*)dest)
- #define **msg\_setserver**(header, value) msg\_setheader((sip\_t \*)header,(char \*)"server",value)
- #define **msg\_getserver**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"server",pos,(header\_t \*\*)dest)
- #define **msg\_setunsupported**(header, value) msg\_setheader((sip\_t \*)header,(char \*)"unsupported",value)
- #define **msg\_getunsupported**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"unsupported",pos,(header\_t \*\*)dest)
- #define **msg\_setwarning**(header, value) msg\_setheader((sip\_t \*)header,(char \*)"warning",value)
- #define **msg\_getwarning**(header, pos, dest) msg\_header\_getbyname(( sip\_t \*)header,"warning",pos,(header\_t \*\*)dest)

## Functions

- int **parser\_init** ()
- int **msg\_init** (sip\_t \*\*sip)
- void **msg\_free** (sip\_t \*sip)
- int **msg\_parse** (sip\_t \*sip, char \*message)
- int **msg\_2char** (sip\_t \*sip, char \*\*dest)
- int **msg\_clone** (sip\_t \*sip, sip\_t \*\*dest)
- int **msg\_force\_update** (sip\_t \*sip)
- char \* **msg\_getreason** (int status\_code)
- void **msg\_setreasonphrase** (sip\_t \*sip, char \*reason)
- char \* **msg\_getreasonphrase** (sip\_t \*sip)
- void **msg\_setstatuscode** (sip\_t \*sip, char \*statuscode)
- char \* **msg\_getstatuscode** (sip\_t \*sip)
- void **msg\_setmethod** (sip\_t \*sip, char \*method)
- char \* **msg\_getmethod** (sip\_t \*sip)
- void **msg\_setversion** (sip\_t \*sip, char \*version)
- char \* **msg\_getversion** (sip\_t \*sip)
- void **msg\_seturi** (sip\_t \*sip, url\_t \*uri)
- url\_t \* **msg\_geturi** (sip\_t \*sip)
- int **msg\_setaccept** (sip\_t \*sip, char \*hvalue)
- int **msg\_getaccept** (sip\_t \*sip, int pos, accept\_t \*\*dest)
- int **msg\_setaccept\_encoding** (sip\_t \*sip, char \*hvalue)
- int **msg\_getaccept\_encoding** (sip\_t \*sip, int pos, accept\_encoding\_t \*\*dest)
- int **msg\_setaccept\_language** (sip\_t \*sip, char \*hvalue)
- int **msg\_getaccept\_language** (sip\_t \*sip, int pos, accept\_language\_t \*\*dest)
- int **msg\_setalert\_info** (sip\_t \*sip, char \*hvalue)

- int **msg\_getalert\_info** (sip\_t \*sip, int pos, **alert\_info\_t** \*\*dest)
- int **msg\_setallow** (sip\_t \*sip, char \*hvalue)
- int **msg\_getallow** (sip\_t \*sip, int pos, **allow\_t** \*\*dest)
- int **msg\_setauthorization** (sip\_t \*sip, char \*hvalue)
- **authorization\_t** \* **msg\_getauthorization** (sip\_t \*sip)
- int **msg\_setcall\_id** (sip\_t \*sip, char \*hvalue)
- **call\_id\_t** \* **msg\_getcall\_id** (sip\_t \*sip)
- int **msg\_setcall\_info** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcall\_info** (sip\_t \*sip, int pos, **call\_info\_t** \*\*dest)
- int **msg\_setcontact** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontact** (sip\_t \*sip, int pos, **contact\_t** \*\*dest)
- int **msg\_setcontent\_disposition** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontent\_disposition** (sip\_t \*sip, int pos, **content\_disposition\_t** \*\*dest)
- int **msg\_setcontent\_encoding** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontent\_encoding** (sip\_t \*sip, int pos, **content\_encoding\_t** \*\*dest)
- int **msg\_setcontent\_length** (sip\_t \*sip, char \*hvalue)
- **content\_length\_t** \* **msg\_getcontent\_length** (sip\_t \*sip)
- int **msg\_setcontent\_type** (sip\_t \*sip, char \*hvalue)
- **content\_type\_t** \* **msg\_getcontent\_type** (sip\_t \*sip)
- int **msg\_setcseq** (sip\_t \*sip, char \*hvalue)
- **cseq\_t** \* **msg\_getcseq** (sip\_t \*sip)
- int **msg\_seterror\_info** (sip\_t \*sip, char \*hvalue)
- int **msg\_geterror\_info** (sip\_t \*sip, int pos, **error\_info\_t** \*\*dest)
- int **msg\_setfrom** (sip\_t \*sip, char \*hvalue)
- **from\_t** \* **msg\_getfrom** (sip\_t \*sip)
- int **msg\_setmime\_version** (sip\_t \*sip, char \*hvalue)
- **mime\_version\_t** \* **msg\_getmime\_version** (sip\_t \*sip)
- int **msg\_setproxy\_authenticate** (sip\_t \*sip, char \*hvalue)
- **proxy\_authenticate\_t** \* **msg\_getproxy\_authenticate** (sip\_t \*sip)
- int **msg\_setproxy\_authorization** (sip\_t \*sip, char \*hvalue)
- int **msg\_getproxy\_authorization** (sip\_t \*sip, int pos, **proxy\_authorization\_t** \*\*dest)
- int **msg\_setrecord\_route** (sip\_t \*sip, char \*hvalue)
- int **msg\_getrecord\_route** (sip\_t \*sip, int pos, **record\_route\_t** \*\*dest)
- int **msg\_setroute** (sip\_t \*sip, char \*hvalue)
- int **msg\_getroute** (sip\_t \*sip, int pos, **route\_t** \*\*dest)
- int **msg\_setto** (sip\_t \*sip, char \*hvalue)
- **to\_t** \* **msg\_getto** (sip\_t \*sip)
- int **msg\_setvia** (sip\_t \*sip, char \*hvalue)
- int **msg\_getvia** (sip\_t \*sip, int pos, **via\_t** \*\*dest)
- int **msg\_setwww\_authenticate** (sip\_t \*sip, char \*hvalue)
- **www\_authenticate\_t** \* **msg\_getwww\_authenticate** (sip\_t \*sip)
- int **msg\_setheader** (sip\_t \*sip, char \*hname, char \*hvalue)
- int **msg\_header\_getbyname** (sip\_t \*sip, char \*hname, int pos, **header\_t** \*\*dest)
- int **msg\_getheader** (sip\_t \*sip, int pos, **header\_t** \*\*dest)
- int **msg\_setbody** (sip\_t \*sip, char \*buf)
- int **msg\_setbody\_mime** (sip\_t \*sip, char \*buf)
- int **msg\_getbody** (sip\_t \*sip, int pos, **body\_t** \*\*dest)
- int **body\_init** (**body\_t** \*\*body)
- void **body\_free** (**body\_t** \*body)
- int **body\_parse** (**body\_t** \*body, char \*buf)

- `int body_parse_mime (body_t *body, char *buf)`
- `int body_2char (body_t *body, char **dest)`
- `void generic_param_setname (generic_param_t *generic_param, char *name)`
- `char * generic_param_getname (generic_param_t *generic_param)`
- `void generic_param_setvalue (generic_param_t *generic_param, char *value)`
- `char * generic_param_getvalue (generic_param_t *generic_param)`
- `int header_init (header_t **header)`
- `void header_free (header_t *header)`
- `int header_2char (header_t *header, char **dest)`
- `char * header_getname (header_t *header)`
- `void header_setname (header_t *header, char *pname)`
- `char * header_getvalue (header_t *header)`
- `void header_setvalue (header_t *header, char *pvalue)`
- `int header_clone (header_t *header, header_t **dest)`
- `int accept_encoding_init (accept_encoding_t **header)`
- `int accept_encoding_parse (accept_encoding_t *header, char *hvalue)`
- `int accept_encoding_2char (accept_encoding_t *header, char **dest)`
- `void accept_encoding_free (accept_encoding_t *header)`
- `int accept_encoding_clone (accept_encoding_t *header, accept_encoding_t **dest)`
- `void accept_encoding_setelement (accept_encoding_t *header, char *value)`
- `char * accept_encoding_getelement (accept_encoding_t *header)`
- `int authorization_init (authorization_t **header)`
- `int authorization_parse (authorization_t *header, char *hvalue)`
- `int authorization_2char (authorization_t *header, char **dest)`
- `void authorization_free (authorization_t *header)`
- `int authorization_clone (authorization_t *header, authorization_t **dest)`
- `char * authorization_getauth_type (authorization_t *header)`
- `void authorization_setauth_type (authorization_t *header, char *value)`
- `char * authorization_getusername (authorization_t *header)`
- `void authorization_setusername (authorization_t *header, char *value)`
- `char * authorization_getrealm (authorization_t *header)`
- `void authorization_setrealm (authorization_t *header, char *value)`
- `char * authorization_getnonce (authorization_t *header)`
- `void authorization_setnonce (authorization_t *header, char *value)`
- `char * authorization_geturi (authorization_t *header)`
- `void authorization_seturi (authorization_t *header, char *value)`
- `char * authorization_getresponse (authorization_t *header)`
- `void authorization_setresponse (authorization_t *header, char *value)`
- `char * authorization_getdigest (authorization_t *header)`
- `void authorization_setdigest (authorization_t *header, char *value)`
- `char * authorization_getalgorithm (authorization_t *header)`
- `void authorization_setalgorithm (authorization_t *header, char *value)`
- `char * authorization_getcnonce (authorization_t *header)`
- `void authorization_setcnonce (authorization_t *header, char *value)`
- `char * authorization_getopaque (authorization_t *header)`
- `void authorization_setopaque (authorization_t *header, char *value)`
- `char * authorization_getmessage_qop (authorization_t *header)`
- `void authorization_setmessage_qop (authorization_t *header, char *value)`
- `char * authorization_getnonce_count (authorization_t *header)`
- `void authorization_setnonce_count (authorization_t *header, char *value)`



- `int call_id_init (call_id_t **header)`
- `void call_id_free (call_id_t *header)`
- `int call_id_parse (call_id_t *header, char *hvalue)`
- `int call_id_2char (call_id_t *header, char **dest)`
- `int call_id_clone (call_id_t *header, call_id_t **dest)`
- `void call_id_setnumber (call_id_t *header, char *value)`
- `char * call_id_getnumber (call_id_t *header)`
- `void call_id_sethost (call_id_t *header, char *value)`
- `char * call_id_gethost (call_id_t *header)`
- `int call_info_init (call_info_t **header)`
- `void call_info_free (call_info_t *header)`
- `int call_info_parse (call_info_t *header, char *hvalue)`
- `int call_info_2char (call_info_t *header, char **dest)`
- `int call_info_clone (call_info_t *header, call_info_t **dest)`
- `char * call_info_geturi (call_info_t *header)`
- `void call_info_seturi (call_info_t *header, char *uri)`
- `int contact_init (contact_t **header)`
- `void contact_free (contact_t *header)`
- `int contact_parse (contact_t *header, char *hvalue)`
- `int contact_2char (contact_t *header, char **dest)`
- `int contact_clone (contact_t *header, contact_t **dest)`
- `int content_disposition_parse (content_disposition_t *header, char *hvalue)`
- `int content_length_init (content_length_t **header)`
- `void content_length_free (content_length_t *header)`
- `int content_length_parse (content_length_t *header, char *hvalue)`
- `int content_length_2char (content_length_t *header, char **dest)`
- `int content_length_clone (content_length_t *header, content_length_t **dest)`
- `int content_type_init (content_type_t **header)`
- `void content_type_free (content_type_t *header)`
- `int content_type_parse (content_type_t *header, char *hvalue)`
- `int content_type_2char (content_type_t *header, char **dest)`
- `int content_type_clone (content_type_t *header, content_type_t **dest)`
- `int cseq_init (cseq_t **header)`
- `void cseq_free (cseq_t *header)`
- `int cseq_parse (cseq_t *header, char *hvalue)`
- `int cseq_2char (cseq_t *header, char **dest)`
- `int cseq_clone (cseq_t *header, cseq_t **dest)`
- `void cseq_setnumber (cseq_t *header, char *value)`
- `char * cseq_getnumber (cseq_t *header)`
- `void cseq_setmethod (cseq_t *header, char *value)`
- `char * cseq_getmethod (cseq_t *header)`
- `int from_init (from_t **header)`
- `void from_free (from_t *header)`
- `int from_parse (from_t *header, char *hvalue)`
- `int from_2char (from_t *header, char **dest)`
- `int from_clone (from_t *header, from_t **dest)`
- `void from_setdisplayname (from_t *header, char *value)`
- `char * from_getdisplayname (from_t *header)`
- `void from_seturl (from_t *header, url_t *url)`
- `url_t * from_geturl (from_t *header)`

- `int from_param_get (from_t *header, int pos, generic_param_t **dest)`
- `int record_route_init (record_route_t **header)`
- `void record_route_free (record_route_t *header)`
- `int record_route_parse (record_route_t *header, char *hvalue)`
- `int record_route_2char (record_route_t *header, char **dest)`
- `int route_init (route_t **header)`
- `void route_free (route_t *header)`
- `int route_parse (route_t *header, char *hvalue)`
- `int route_2char (route_t *header, char **dest)`
- `int to_init (to_t **header)`
- `void to_free (to_t *header)`
- `int to_parse (to_t *header, char *hvalue)`
- `int to_2char (to_t *header, char **dest)`
- `int to_clone (to_t *header, to_t **dest)`
- `int via_init (via_t **header)`
- `void via_free (via_t *header)`
- `int via_parse (via_t *header, char *hvalue)`
- `int via_2char (via_t *header, char **dest)`
- `int via_clone (via_t *header, via_t **dest)`
- `void via_setversion (via_t *header, char *value)`
- `char * via_getversion (via_t *header)`
- `void via_setprotocol (via_t *header, char *value)`
- `char * via_getprotocol (via_t *header)`
- `void via_sethost (via_t *header, char *value)`
- `char * via_gethost (via_t *header)`
- `void via_setport (via_t *header, char *value)`
- `char * via_getport (via_t *header)`
- `void via_setcomment (via_t *header, char *value)`
- `char * via_getcomment (via_t *header)`
- `int www_authenticate_init (www_authenticate_t **header)`
- `int www_authenticate_parse (www_authenticate_t *header, char *hvalue)`
- `int www_authenticate_2char (www_authenticate_t *header, char **dest)`
- `void www_authenticate_free (www_authenticate_t *header)`
- `int www_authenticate_clone (www_authenticate_t *header, www_authenticate_t **dest)`
- `char * www_authenticate_getauth_type (www_authenticate_t *header)`
- `void www_authenticate_setauth_type (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getrealm (www_authenticate_t *header)`
- `void www_authenticate_setrealm (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getdomain (www_authenticate_t *header)`
- `void www_authenticate_setdomain (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getnonce (www_authenticate_t *header)`
- `void www_authenticate_setnonce (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getopaque (www_authenticate_t *header)`
- `void www_authenticate_setopaque (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getstale (www_authenticate_t *header)`
- `void www_authenticate_setstale (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getalgorithm (www_authenticate_t *header)`
- `void www_authenticate_setalgorithm (www_authenticate_t *header, char *value)`
- `char * www_authenticate_getqop_options (www_authenticate_t *header)`
- `void www_authenticate_setqop_options (www_authenticate_t *header, char *value)`

### 4.8.1 Define Documentation

#### 4.8.1.1 `#define accept_2char(header, dest) content_type_2char(header, dest)`

Get a string representation of an Accept element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

#### 4.8.1.2 `#define accept_clone(header, dest) content_type_clone(header, dest)`

Clone an Accept element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

#### 4.8.1.3 `#define accept_encoding_param_add(header, name, value)` `generic_param_add((header) → gen_params,name,value)`

Allocate and Add a header parameter in an Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*name* The token name for the new parameter.

*value* The token value for the new parameter.

#### 4.8.1.4 `#define accept_encoding_param_get(header, pos, dest)` `generic_param_get((header) → gen_params, pos, dest)`

Get a header parameter from an Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

#### 4.8.1.5 `#define accept_encoding_param_getbyname(header, name, dest)` `generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in an Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.6 #define accept\_free(header) content\_type\_free(header)**

Free an Accept element.

**Parameters:**

*header* The element to work on.

**4.8.1.7 #define accept\_init(header) content\_type\_init(header)**

Allocate an Accept element.

**Parameters:**

*header* The element to work on.

**4.8.1.8 #define accept\_language\_2char(header, dest) accept\_encoding\_2char(header, dest)**

Get a string representation of an Accept-Language element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.9 #define accept\_language\_clone(header, dest) accept\_encoding\_clone(header, dest)**

Clone an Accept-Language element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.10 #define accept\_language\_free(header) accept\_encoding\_free(header)**

Free an Accept-Language element.

**Parameters:**

*header* The element to work on.

**4.8.1.11 #define accept\_language\_getelement(header) accept\_encoding\_getelement(header)**

Get the value of an Accept-Language element.

**Parameters:**

*header* The element to work on.

**4.8.1.12** `#define accept_language_init(header) accept_encoding_init(header)`

Allocate an Accept-Language element.

**Parameters:**

*header* The element to work on.

**4.8.1.13** `#define accept_language_param_add(header, name, value)  
generic_param_add((header) → gen_params,name,value)`

Allocate and add a generic parameter element in an Accept-Language element.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.14** `#define accept_language_param_get(header, pos, dest)  
generic_param_get((header) → gen_params, pos,dest)`

Get a header parameter from an Accept-Language element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.15** `#define accept_language_param_getbyname(header, name, dest)  
generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a Accept-Language element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.16** `#define accept_language_parse(header, hvalue) accept_encoding_  
parse(header, hvalue)`

Parse an Accept-Language element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.17** `#define accept_language_setelement(header, value)`  
`accept_encoding_setelement(header, value)`

Set the value of an Accept-Language element.

**Parameters:**

*header* The element to work on.

*value* The value to set.

**4.8.1.18** `#define accept_param_add(header, name, value)`  
`generic_param_add((header) → gen_params,name,value)`

Allocate and add a header parameter in an Accept element.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.19** `#define accept_param_get(header, pos, dest)` `generic_param_get((header)`  
`→ gen_params, pos, dest)`

Get a header parameter from an Accept element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.20** `#define accept_param_getbyname(header, name, dest)`  
`generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in an Accept element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.21** `#define accept_parse(header, hvalue)` `content_type_parse(header, hvalue)`

Parse an Accept element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.22** `#define alert_info_2char(header, dest) call_info_2char(header,dest)`

Get a string representation of a Alert-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.23** `#define alert_info_clone(header, dest) call_info_clone(header, dest)`

Clone a Alert-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.24** `#define alert_info_free(header) call_info_free(header)`

Free a Alert-Info element.

**Parameters:**

*header* The element to work on.

**4.8.1.25** `#define alert_info_geturi(header) call_info_geturi(header)`

Get uri from an Alert-Info element.

**Parameters:**

*header* The element to work on.

**4.8.1.26** `#define alert_info_init(header) call_info_init(header)`

Allocate a Alert-Info element.

**Parameters:**

*header* The element to work on.

**4.8.1.27** `#define alert_info_parse(header, hvalue) call_info_parse(header, hvalue)`

Parse a Alert-Info element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.28** `#define alert_info_seturi(header, uri) call_info_seturi(header, uri)`

Set the uri of an Alert-Info element.

**Parameters:**

*header* The element to work on.

*uri* The value of the new parameter.

**4.8.1.29** `#define allow_2char(header, dest) content_length_2char(header, dest)`

Get a string representation of a Allow element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.30** `#define allow_clone(header, dest) content_length_clone(header, dest)`

Clone a Allow element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.31** `#define allow_free(header) content_length_free(header)`

Free a Allow element.

**Parameters:**

*header* The element to work on.

**4.8.1.32** `#define allow_init(header) content_length_init(header)`

Allocate a Allow element.

**Parameters:**

*header* The element to work on.

**4.8.1.33** `#define allow_parse(header, hvalue) content_length_parse(header, hvalue)`

Parse a Allow element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.



**4.8.1.34** `#define contact_getdisplayname(header) from_getdisplayname((from_t*)header)`

Get the displayname from a Contact header.

**Parameters:**

*header* The element to work on.

**4.8.1.35** `#define contact_geturl(header) from_geturl((from_t*)header)`

Get the url from a Contact header.

**Parameters:**

*header* The element to work on.

**4.8.1.36** `#define contact_param_add(header, name, value)  
generic_param_add((header) → gen_params, name,value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.37** `#define contact_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`

Get a header parameter from a Contact element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.38** `#define contact_param_getbyname(header, name, dest)  
generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a Contact element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.39** `#define contact_setdisplayname(header, value)`  
`from_setdisplayname((from_t*)header, value)`

Set the displayname in the Contact element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.1.40** `#define contact_seturl(header, url) from_seturl((from_t*)header,url)`

Set the url in the Contact element.

**Parameters:**

*header* The element to work on.

*url* The value of the element.

**4.8.1.41** `#define content_disposition_2char(header, dest)`  
`call_info_2char(header,dest)`

Get a string representation of a Content-Disposition element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.42** `#define content_disposition_clone(header, dest) call_info_clone(header,`  
`dest)`

Clone a Content-Disposition element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.43** `#define content_disposition_free(header) call_info_free(header)`

Free a Content-Disposition element.

**Parameters:**

*header* The element to work on.

**4.8.1.44** `#define content_disposition_gettype(header) call_info_geturi(header)`

Get the type from a Content-Disposition header.

**Parameters:**

*header* The element to work on.

**4.8.1.45** `#define content_disposition_init(header) call_info_init(header)`

Allocate a Content-Disposition element.

**Parameters:**

*header* The element to work on.

**4.8.1.46** `#define content_disposition_settype(header, value) call_info_seturi(header, value)`

Set the type in the Content-Disposition element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.1.47** `#define content_encoding_2char(header, dest) content_length_2char(header, dest)`

Get a string representation of a Content-Encoding element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.48** `#define content_encoding_clone(header, dest) content_length_clone(header, dest)`

Clone a Content-Encoding element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.49** `#define content_encoding_free(header) content_length_free(header)`

Free a Content-Encoding element.

**Parameters:**

*header* The element to work on.

**4.8.1.50** `#define content_encoding_init(header) content_length_init(header)`

Allocate a Content-Encoding element.

**Parameters:**

*header* The element to work on.

**4.8.1.51** `#define content_encoding_parse(header, hvalue)  
content_length_parse(header, hvalue)`

Parse a Content-Encoding element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.52** `#define content_type_param_add(header, name, value)  
generic_param_add((header) → gen_params,name,value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.53** `#define content_type_param_get(header, pos, dest)  
generic_param_get((header) → gen_params, pos,dest)`

Get a header parameter from a Content-Type element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.54** `#define content_type_param_getbyname(header, name, dest)  
generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a Content-Type element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.55** `#define error_info_2char(header, dest) call_info_2char(header,dest)`

Get a string representation of a Error-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.56** `#define error_info_clone(header, dest) call_info_clone(header, dest)`

Clone a Error-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.57** `#define error_info_free(header) call_info_free(header)`

Free a Error-Info element.

**Parameters:**

*header* The element to work on.

**4.8.1.58** `#define error_info_geturi(header) call_info_geturi(header)`

Get the uri from a Error-Info header.

**Parameters:**

*header* The element to work on.

**4.8.1.59** `#define error_info_init(header) call_info_init(header)`

Allocate a Error-Info element.

**Parameters:**

*header* The element to work on.

**4.8.1.60** `#define error_info_parse(header, hvalue) call_info_parse(header, hvalue)`

Parse a Error-Info element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.61** `#define error_info_seturi(header, uri) call_info_seturi(header, uri)`

Set the uri in the Error-Info element.

**Parameters:**

*header* The element to work on.

*uri* The uri of the element.

**4.8.1.62** `#define from_get_tag(header, dest) generic_param_getbyname((header) → gen_params, "tag", dest)`

Find the tag parameter in a From element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the element found.

**4.8.1.63** `#define from_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.64** `#define from_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`

Find a header parameter in a From element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.65** `#define from_set_tag(header, value) generic_param_add((header) → gen_params, sgetcopy("tag"), value)`

Allocate and add a tag parameter element in a Contact element.

**Parameters:**

*header* The element to work on.

*value* The token value.

**4.8.1.66** `#define generic_param_add(LIST, NAME, VALUE) url_param_add(LIST, NAME, VALUE)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*LIST* The list of generic parameter element to work on.

*NAME* The token name.

*VALUE* The token value.

**4.8.1.67 #define generic\_param\_clone(GP, DEST) url\_param\_clone(GP,DEST)**

Clone a generic parameter element.

**Parameters:**

**GP** The element to work on.

**DEST** The resulting new allocated buffer.

**4.8.1.68 #define generic\_param\_free(GP) url\_param\_free(GP)**

Free a generic parameter element.

**Parameters:**

**GP** The element to work on.

**4.8.1.69 #define generic\_param\_getbyname(LIST, NAME, DEST)  
url\_param\_getbyname(LIST,NAME,DEST)**

Find in a generic parameter element in a list.

**Parameters:**

**LIST** The list of generic parameter element to work on.

**NAME** The name of the parameter element to find.

**DEST** A pointer on the element found.

**4.8.1.70 #define generic\_param\_init(GP) url\_param\_init(GP)**

Allocate a generic parameter element.

**Parameters:**

**GP** The element to work on.

**4.8.1.71 #define generic\_param\_set(GP, NAME, VALUE) url\_param\_set(GP,  
NAME, VALUE)**

Set values of a generic parameter element.

**Parameters:**

**GP** The element to work on.

**NAME** The token name.

**VALUE** The token value.

**4.8.1.72 #define mime\_version\_2char(header, dest) content\_length\_2char(header,  
dest)**

Get a string representation of a Mime-Version element.

**Parameters:**

**header** The element to work on.

**dest** A pointer on the new allocated string.

**4.8.1.73** `#define mime_version_clone(header, dest) content_length_clone(header, dest)`

Clone a Mime-Version element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.74** `#define mime_version_free(header) content_length_free(header)`

Free a Mime-Version element.

**Parameters:**

*header* The element to work on.

**4.8.1.75** `#define mime_version_init(header) content_length_init(header)`

Allocate a Mime-Version element.

**Parameters:**

*header* The element to work on.

**4.8.1.76** `#define mime_version_parse(header, hvalue) content_length_parse(header, hvalue)`

Parse a Mime-Version element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.77** `#define msg_getcontent_language(header, pos, dest)  
msg_header_getbyname(( sip_t *)header,"content-language",pos,(header_t  
**)dest)`

Find a Content-Language header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.



```
4.8.1.78 #define msg_getdate(header, pos, dest) msg_header_getbyname(( sip_t
*)header,"date",pos,(header_t **)dest)
```

Find a Date header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.79 #define msg_getencryption(header, pos, dest) msg_header_getbyname((
sip_t *)header,"encryption",pos,(header_t **)dest)
```

Find an Encryption header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.80 #define msg_getexpires(header, pos, dest) msg_header_getbyname(( sip_t
*)header,"expires",pos,(header_t **)dest)
```

Find a Expires header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.81 #define msg_getin_reply_to(header, pos, dest) msg_header_getbyname((
sip_t *)header,"in-reply-to",pos,(header_t **)dest)
```

Find a In-Reply-To header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.82 #define msg_getmax_forward(header, pos, dest) msg_header_getbyname((
sip_t *)header,"max-forward",pos,(header_t **)dest)
```

Find a Max-Forward header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.83** `#define msg_getorganization(header, pos, dest) msg_header_getbyname(( sip_t *)header,"organization",pos,(header_t **)dest)`

Find an Organization header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.84** `#define msg_getpriority(header, pos, dest) msg_header_getbyname(( sip_t *)header,"priority",pos,(header_t **)dest)`

Find a Priority header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.85** `#define msg_getproxy_require(header, pos, dest) msg_header_getbyname(( sip_t *)header,"proxy-require",pos,(header_t **)dest)`

Find a Proxy-Require header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.86** `#define msg_getrequire(header, pos, dest) msg_header_getbyname(( sip_t *)header,"require",pos,(header_t **)dest)`

Find a Require header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.87** `#define msg_getresponse_key(header, pos, dest) msg_header_getbyname(( sip_t *)header,"response-key",pos,(header_t **)dest)`

Find a Response-Key header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.88 #define msg_getretry_after(header, pos, dest) msg_header_getbyname((  
sip_t *)header,"retry-after",pos,(header_t **)dest)
```

Find a Retry-After header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.89 #define msg_getserver(header, pos, dest) msg_header_getbyname(( sip_t  
*)header,"server",pos,(header_t **)dest)
```

Find a Server header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.90 #define msg_getsubject(header, pos, dest) msg_header_getbyname(( sip_t  
*)header,"subject",pos,(header_t **)dest)
```

Find a Subject header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.91 #define msg_getsupported(header, pos, dest) msg_header_getbyname((  
sip_t *)header,"supported",pos,(header_t **)dest)
```

Find a Supported header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

```
4.8.1.92 #define msg_gettimestamp(header, pos, dest) msg_header_getbyname((  
sip_t *)header,"timestamp",pos,(header_t **)dest)
```

Find a Timestamp header.

**Parameters:**

*header* The element to work on.

*pos* The index of the header in the list of unknown header.

*dest* A pointer on the element found.

**4.8.1.93** `#define msg_getunsupported(header, pos, dest) msg_header_getbyname((  
sip_t *)header,"unsupported",pos,(header_t **)dest)`

Find a Unsupported header.

**Parameters:**

*header* The element to work on.  
*pos* The index of the header in the list of unknown header.  
*dest* A pointer on the element found.

**4.8.1.94** `#define msg_getuser_agent(header, pos, dest) msg_header_getbyname((  
sip_t *)header,"user-agent",pos,(header_t **)dest)`

Find a User-Agent header.

**Parameters:**

*header* The element to work on.  
*pos* The index of the header in the list of unknown header.  
*dest* A pointer on the element found.

**4.8.1.95** `#define msg_getwarning(header, pos, dest) msg_header_getbyname(( sip_t  
*)header,"warning",pos,(header_t **)dest)`

Find a Warning header.

**Parameters:**

*header* The element to work on.  
*pos* The index of the header in the list of unknown header.  
*dest* A pointer on the element found.

**4.8.1.96** `#define MSG_IS_ACK(msg)`

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strtlne->sipmethod,"ACK",3))
```

Test if the message is an ACK REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.97** `#define MSG_IS_BYE(msg)`

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strtlne->sipmethod,"BYE",3))
```

Test if the message is a BYE REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.98 #define MSG\_IS\_CANCEL(msg)**

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"CANCEL",6))
```

Test if the message is a CANCEL REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.99 #define MSG\_IS\_INFO(msg)**

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"INFO",4))
```

Test if the message is an INFO REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.100 #define MSG\_IS\_INVITE(msg)**

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"INVITE",6))
```

Test if the message is an INVITE REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.101 #define MSG\_IS\_NOTIFY(msg)**

**Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"NOTIFY",6))
```

Test if the message is a NOTIFY REQUEST

**Parameters:**

*msg* the SIP message.

**4.8.1.102 #define MSG\_IS\_OPTIONS(msg)****Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"OPTIONS",7))
```

Test if the message is an OPTIONS REQUEST

**Parameters:***msg* the SIP message.**4.8.1.103 #define MSG\_IS\_PRACK(msg)****Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"PRACK",5))
```

Test if the message is a PRACK REQUEST (!! PRACK IS NOT SUPPORTED by the fsm!!)

**Parameters:***msg* the SIP message.**4.8.1.104 #define MSG\_IS\_REGISTER(msg)****Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strcmp((msg)->strline->sipmethod,"REGISTER",8))
```

Test if the message is a REGISTER REQUEST

**Parameters:***msg* the SIP message.**4.8.1.105 #define MSG\_IS\_REQUEST(msg) ((msg) → strline → statuscode==NULL)**

Test if the message is a SIP REQUEST

**Parameters:***msg* the SIP message.**4.8.1.106 #define MSG\_IS\_RESPONSE(msg) ((msg) → strline → statuscode!=NULL)**

Test if the message is a SIP RESPONSE

**Parameters:***msg* the SIP message.

**4.8.1.107 #define MSG\_IS\_RESPONSEFOR(msg, requestname)**

**Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strcmp((msg)->cseq->method,requestname))
```

Test if the message is a response for a REQUEST of certain type

**Parameters:**

*msg* the SIP message.

*requestname* the method name to match.

**4.8.1.108 #define MSG\_IS\_STATUS\_1XX(msg)**

**Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"1",1))
```

Test if the message is a response with status between 100 and 199

**Parameters:**

*msg* the SIP message.

**4.8.1.109 #define MSG\_IS\_STATUS\_2XX(msg)**

**Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"2",1))
```

Test if the message is a response with status between 200 and 299

**Parameters:**

*msg* the SIP message.

**4.8.1.110 #define MSG\_IS\_STATUS\_3XX(msg)**

**Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"3",1))
```

Test if the message is a response with status between 300 and 399

**Parameters:**

*msg* the SIP message.

**4.8.1.111 #define MSG\_IS\_STATUS\_4XX(msg)****Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"4",1))
```

Test if the message is a response with status between 400 and 499

**Parameters:**

*msg* the SIP message.

**4.8.1.112 #define MSG\_IS\_STATUS\_5XX(msg)****Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"5",1))
```

Test if the message is a response with status between 500 and 599

**Parameters:**

*msg* the SIP message.

**4.8.1.113 #define MSG\_IS\_STATUS\_6XX(msg)****Value:**

```
(MSG_IS_RESPONSE(msg) && \
    0==strncmp((msg)->strline->statuscode,"6",1))
```

Test if the message is a response with status between 600 and 699

**Parameters:**

*msg* the SIP message.

**4.8.1.114 #define MSG\_IS\_SUBSCRIBE(msg)****Value:**

```
(MSG_IS_REQUEST(msg) && \
    0==strncmp((msg)->strline->sipmethod,"SUBSCRIBE",9))
```

Test if the message is a SUBSCRIBE REQUEST

**Parameters:**

*msg* the SIP message.



**4.8.1.115** `#define msg_setcontent_language(header, value) msg_setheader((sip_t *)header,(char *)"content-language",value)`

Allocate and Add a new Content-Language header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.116** `#define msg_setdate(header, value) msg_setheader((sip_t *)header,(char *)"date",value)`

Allocate and Add a new Date header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.117** `#define msg_setencryption(header, value) msg_setheader((sip_t *)header,(char *)"encryption",value)`

Allocate and Add a new Encryption header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.118** `#define msg_setexpires(header, value) msg_setheader((sip_t *)header,(char *)"expires",value)`

Allocate and Add a new Expires header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.119** `#define msg_setin_reply_to(header, value) msg_setheader((sip_t *)header,(char *)"in-reply-to",value)`

Allocate and Add a new In-Reply-To header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.120** `#define msg_setmax_forward(header, value) msg_setheader((sip_t *)header,(char *)"max-forward",value)`

Allocate and Add a new Max-Forward header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.121** `#define msg_setorganization(header, value) msg_setheader((sip_t *)header,(char *)"organization",value)`

Allocate and Add a new Organization header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.122** `#define msg_setpriority(header, value) msg_setheader((sip_t *)header,(char *)"priority",value)`

Allocate and Add a new Priority header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.123** `#define msg_setproxy_require(header, value) msg_setheader((sip_t *)header,(char *)"proxy-require",value)`

Allocate and Add a new Proxy-Require header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.124** `#define msg_setrequire(header, value) msg_setheader((sip_t *)header,(char *)"require",value)`

Allocate and Add a new Require header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.125** `#define msg_setresponse_key(header, value) msg_setheader((sip_t *)header,(char *)"response-key",value)`

Allocate and Add a new Response-Key header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.126** `#define msg_setretry_after(header, value) msg_setheader((sip_t *)header,(char *)"retry-after",value)`

Allocate and Add a new Retry-After header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.127** `#define msg_setserver(header, value) msg_setheader((sip_t *)header,(char *)"server",value)`

Allocate and Add a new Server header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.128** `#define msg_setsubject(header, value) msg_setheader((sip_t *)header,(char *)"subject",value)`

Allocate and Add a new Subject header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.129** `#define msg_setsupported(header, value) msg_setheader((sip_t *)header,(char *)"supported",value)`

Allocate and Add a new Supported header.

**Parameters:**

*header* The element to work on.

*value* the value of the new header.

**4.8.1.130** `#define msg_settimestamp(header, value) msg_setheader((sip_t *)header,(char *)"timestamp",value)`

Allocate and Add a new Timestamp header.

**Parameters:**

*header* The element to work on.  
*value* the value of the new header.

**4.8.1.131** `#define msg_setunsupported(header, value) msg_setheader((sip_t *)header,(char *)"unsupported",value)`

Allocate and Add a new Unsupported header.

**Parameters:**

*header* The element to work on.  
*value* the value of the new header.

**4.8.1.132** `#define msg_setuser_agent(header, value) msg_setheader((sip_t *)header,(char *)"user-agent",value)`

Allocate and Add a new User-Agent header.

**Parameters:**

*header* The element to work on.  
*value* the value of the new header.

**4.8.1.133** `#define msg_setwarning(header, value) msg_setheader((sip_t *)header,(char *)"warning",value)`

Allocate and Add a new Warning header.

**Parameters:**

*header* The element to work on.  
*value* the value of the new header.

**4.8.1.134** `#define MSG_TEST_CODE(msg, code)`

**Value:**

```
(MSG_IS_RESPONSE(msg) && \
    code==(int)satoi((msg)->strline->statuscode))
```

Test if the message is a response with a status set to the code value.

**Parameters:**

*msg* the SIP message.  
*code* the status code.

**4.8.1.135** `#define proxy_authenticate_2char(header, dest)`  
`www_authenticate_2char(header, dest)`

Get a string representation of a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.136** `#define proxy_authenticate_clone(header, dest)`  
`www_authenticate_clone(header, dest)`

Clone a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.137** `#define proxy_authenticate_free(header) www_authenticate_free(header)`

Free a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.138** `#define proxy_authenticate_getalgorithm(header)`  
`www_authenticate_getalgorithm(header)`

Get value of the algorithm parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.139** `#define proxy_authenticate_getauth_type(header)`  
`www_authenticate_getauth_type(header)`

Get value of the auth\_type parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.140** `#define proxy_authenticate_getdomain(header)`  
`www_authenticate_getdomain(header)`

Get value of the domain parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.141** `#define proxy_authenticate_getnonce(header) www_authenticate_-  
getnonce(header)`

Get value of the nonce parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.142** `#define proxy_authenticate_getopaque(header) www_authenticate_-  
getopaque(header)`

Get value of the opaque parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.143** `#define proxy_authenticate_getqop_options(header)  
www_authenticate_getqop_options(header)`

Get value of the qop\_options parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.144** `#define proxy_authenticate_getrealm(header) www_authenticate_-  
getrealm(header)`

Get value of the realm parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.145** `#define proxy_authenticate_getstale(header) www_authenticate_-  
getstale(header)`

Get value of the stale parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.146** `#define proxy_authenticate_init(header) www_authenticate_init(header)`

Allocate a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.147** `#define proxy_authenticate_parse(header, hvalue)`  
`www_authenticate_parse(header, hvalue)`

Parse a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.148** `#define proxy_authenticate_setalgorithm(header, value)`  
`www_authenticate_setalgorithm(header, value)`

Add the algorithm parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.149** `#define proxy_authenticate_setalgorithm_MD5(header)`  
`www_authenticate_setalgorithm(header, sgetcopy("MD5"))`

Add the algorithm parameter set to "MD5" in a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.150** `#define proxy_authenticate_setauth_type(header, value)`  
`www_authenticate_setauth_type(header, value)`

Add the auth\_type parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.151** `#define proxy_authenticate_setdomain(header, value)`  
`www_authenticate_setdomain(header, value)`

Add the domain parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.152** `#define proxy_authenticate_setnonce(header, value)`  
`www_authenticate_setnonce(header, value)`

Add the nonce parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.153** `#define proxy_authenticate_setopaque(header, value)`  
`www_authenticate_setopaque(header, value)`

Add the opaque parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.154** `#define proxy_authenticate_setqop_options(header, value)`  
`www_authenticate_setqop_options(header,value)`

Add the qop-options parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.155** `#define proxy_authenticate_setrealm(header, value)`  
`www_authenticate_setrealm(header, value)`

Add the realm parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.156** `#define proxy_authenticate_setstale(header, value)`  
`www_authenticate_setstale(header, value)`

Add the stale parameter from a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.



**4.8.1.157** `#define proxy_authenticate_setstale_false(header)  
www_authenticate_setstale(header,sgetcopy("false"))`

Add a stale parameter set to "false" in a Proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.158** `#define proxy_authenticate_setstale_true(header)  
www_authenticate_setstale(header,sgetcopy("true"))`

Add a stale parameter set to "true" in a proxy-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.159** `#define proxy_authorization_2char(header, dest)  
authorization_2char(header, dest)`

Get a string representation of a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.1.160** `#define proxy_authorization_clone(header, dest)  
authorization_clone(header, dest)`

Clone a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.161** `#define proxy_authorization_free(header) authorization_free(header)`

Free a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.162** `#define proxy_authorization_getalgorithm(header)  
authorization_getalgorithm(header)`

Get value of the algorithm parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.163** `#define proxy_authorization_getauth_type(header)`  
`authorization_getauth_type(header)`

Get value of the `auth_type` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.164** `#define proxy_authorization_getcnonce(header)`  
`authorization_getcnonce(header)`

Get value of the `cnonce` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.165** `#define proxy_authorization_getdigest(header)` `authorization_-`  
`getdigest(header)`

Get value of the `digest` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.166** `#define proxy_authorization_getmessage_qop(header)`  
`authorization_getmessage_qop(header)`

Get value of the `message_qop` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.167** `#define proxy_authorization_getnonce(header)` `authorization_-`  
`getnonce(header)`

Get value of the `nonce` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.168** `#define proxy_authorization_getnonce_count(header)`  
`authorization_getnonce_count(header)`

Get value of the `nonce_count` parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.169** `#define proxy_authorization_getopaque(header)  
authorization_getopaque(header)`

Get value of the opaque parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.170** `#define proxy_authorization_getrealm(header) authorization_  
getrealm(header)`

Get value of the realm parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.171** `#define proxy_authorization_getresponse(header)  
authorization_getresponse(header)`

Get value of the response parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.172** `#define proxy_authorization_geturi(header) authorization_geturi(header)`

Get value of the uri parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.173** `#define proxy_authorization_getusername(header)  
authorization_getusername(header)`

Get value of the username parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.174** `#define proxy_authorization_init(header) authorization_init(header)`

Allocate a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.1.175** `#define proxy_authorization_parse(header, hvalue)`  
`authorization_parse(header, hvalue)`

Parse a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.1.176** `#define proxy_authorization_setalgorithm(header, value)`  
`authorization_setalgorithm(header,value)`

Add the algorithm parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.177** `#define proxy_authorization_setauth_type(header, value)`  
`authorization_setauth_type(header, value)`

Add the auth\_type parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.178** `#define proxy_authorization_setcnonce(header, value)`  
`authorization_setcnonce(header, value)`

Add the cnonce parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.179** `#define proxy_authorization_setdigest(header, value)`  
`authorization_setdigest(header, value)`

Add the digest parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.180** `#define proxy_authorization_setmessage_qop(header, value)`  
`authorization_setmessage_qop(header, value)`

Add the message\_qop parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.181** `#define proxy_authorization_setnonce(header, value)`  
`authorization_setnonce(header, value)`

Add the nonce parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.182** `#define proxy_authorization_setnonce_count(header, value)`  
`authorization_setnonce_count(header, value)`

Add the nonce\_count parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.183** `#define proxy_authorization_setopaque(header, value)`  
`authorization_setopaque(header, value)`

Add the opaque parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.184** `#define proxy_authorization_setrealm(header, value)`  
`authorization_setrealm(header, value)`

Add the realm parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.185** `#define proxy_authorization_setresponse(header, value)`  
`authorization_setresponse(header, value)`

Add the response parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.186** `#define proxy_authorization_seturi(header, value)`  
`authorization_seturi(header, value)`

Add the uri parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.187** `#define proxy_authorization_setusername(header, value)`  
`authorization_setusername(header, value)`

Add the username parameter from a Proxy-Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.1.188** `#define record_route_clone(header, dest) from_clone(header,dest)`

Clone a Record-Route element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.1.189** `#define record_route_geturl(header) from_geturl((from_t*)header)`

Get the url from a Record-Route header.

**Parameters:**

*header* The element to work on.

**4.8.1.190** `#define record_route_param_add(header, name, value)`  
`generic_param_add((header) → gen_params,name,value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.  
*name* The token name.  
*value* The token value.

**4.8.1.191** `#define record_route_param_get(header, pos, dest)  
from_param_get((from_t*)header,pos,dest)`

Get a header parameter from a Record-Route element.

**Parameters:**

*header* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the element found.

**4.8.1.192** `#define record_route_param_getbyname(header, name, dest)  
generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a Record-Route element.

**Parameters:**

*header* The element to work on.  
*name* The token name to search.  
*dest* A pointer on the element found.

**4.8.1.193** `#define record_route_seturl(header, url) from_seturl((from_t*)header,url)`

Set the url in the Record-Route element.

**Parameters:**

*header* The element to work on.  
*url* The value of the element.

**4.8.1.194** `#define route_clone(header, dest) from_clone(header,dest)`

Clone a Route element.

**Parameters:**

*header* The element to work on.  
*dest* A pointer on the copy of the element.

**4.8.1.195** `#define route_geturl(header) from_geturl((from_t*)header)`

Get the url from a Route header.

**Parameters:**

*header* The element to work on.

**4.8.1.196** `#define route_param_add(header, name, value)`  
`generic_param_add((header) → gen_params,name,value)`

Allocate and add a generic parameter element in a Route element.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.197** `#define route_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`

Get a header parameter from a Route element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.198** `#define route_param_getbyname(header, name, dest)`  
`generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a Route element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.199** `#define route_seturl(header, url) from_seturl((from_t*)header,url)`

Set the url in the Route element.

**Parameters:**

*header* The element to work on.

*url* The value of the element.

**4.8.1.200** `#define to_get_tag(header, dest) generic_param_getbyname((header) → gen_params, "tag",dest)`

Find a tag parameter in a To element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the element found.



**4.8.1.201** `#define to_getdisplayname(header) from_getdisplayname((from_t*)header)`

Get the displayname from a To header.

**Parameters:**

*header* The element to work on.

**4.8.1.202** `#define to_geturl(header) from_geturl((from_t*)header)`

Get the url from a To header.

**Parameters:**

*header* The element to work on.

**4.8.1.203** `#define to_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.204** `#define to_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`

Get a header parameter from a To element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

**4.8.1.205** `#define to_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`

Find a header parameter in a To element.

**Parameters:**

*header* The element to work on.

*name* The token name to search.

*dest* A pointer on the element found.

**4.8.1.206** `#define to_set_tag(header, value) generic_param_add((header) → gen_params, sgetcopy("tag"),value)`

Allocate and add a tag parameter element in a list.

**Parameters:**

*header* The element to work on.

*value* The token value.

**4.8.1.207** `#define to_setdisplayname(header, value) from_setdisplayname((from_t*)header,value)`

Set the displayname in the To element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.1.208** `#define to_seturl(header, url) from_seturl((from_t*)header,url)`

Set the url in the To element.

**Parameters:**

*header* The element to work on.

*url* The value of the element.

**4.8.1.209** `#define via_param_add(header, name, value) generic_param_add((header) → via_params,name,value)`

Allocate and add a generic parameter element in a list.

**Parameters:**

*header* The element to work on.

*name* The token name.

*value* The token value.

**4.8.1.210** `#define via_param_get(header, pos, dest) generic_param_get(header,pos,dest)`

Get a header parameter from a Via element.

**Parameters:**

*header* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the element found.

```
4.8.1.211 #define via_param_getbyname(header, name, dest)
          generic_param_getbyname((header) → via_params,name,dest)
```

Find a header parameter in a Via element.

**Parameters:**

*header* The element to work on.  
*name* The token name to search.  
*dest* A pointer on the element found.

```
4.8.1.212 #define via_set_branch(header, value) generic_param_add((header) →
          via_params,sgetcopy("branch"),value)
```

Allocate and add a branch parameter element in a list.

**Parameters:**

*header* The element to work on.  
*value* The token value.

```
4.8.1.213 #define via_set_hidden(header) generic_param_add((header) →
          via_params,sgetcopy("hidden"),NULL)
```

Allocate and add a hidden parameter element in a list.

**Parameters:**

*header* The element to work on.

```
4.8.1.214 #define via_set_maddr(header, value) generic_param_add((header) →
          via_params,sgetcopy("maddr"),value)
```

Allocate and add a maddr parameter element in a list.

**Parameters:**

*header* The element to work on.  
*value* The token value.

```
4.8.1.215 #define via_set_received(header, value) generic_param_add((header) →
          via_params,sgetcopy("received"),value)
```

Allocate and add a received parameter element in a list.

**Parameters:**

*header* The element to work on.  
*value* The token value.

**4.8.1.216** `#define via_set_ttl(header, value) generic_param_add((header) →  
via_params,sgetcopy("ttl"),value)`

Allocate and add a ttl parameter element in a list.

**Parameters:**

*header* The element to work on.

*value* The token value.

**4.8.1.217** `#define www_authenticate_setalgorithm_MD5(header)  
www_authenticate_setalgorithm(header,sgetcopy("MD5"))`

Add the algorithm parameter set to "MD5" in a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.218** `#define www_authenticate_setstale_false(header)  
www_authenticate_setstale(header,sgetcopy("false"))`

Add a stale parameter set to "false" in a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.1.219** `#define www_authenticate_setstale_true(header)  
www_authenticate_setstale(header,sgetcopy("true"))`

Add a stale parameter set to "true" in a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

## 4.8.2 Function Documentation

**4.8.2.1** `int accept_encoding_2char (accept_encoding_t * header, char ** dest)`

Get a string representation of a Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.2** `int accept_encoding_clone (accept_encoding_t * header, accept_encoding_t **  
dest)`

Clone a Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.3 void accept\_encoding\_free (accept\_encoding\_t \* *header*)**

Free a Accept-Encoding element.

**Parameters:**

*header* The element to work on.

**4.8.2.4 char\* accept\_encoding\_getelement (accept\_encoding\_t \* *header*)**

Get the value of an Accept-Encoding element.

**Parameters:**

*header* The element to work on.

**4.8.2.5 int accept\_encoding\_init (accept\_encoding\_t \*\* *header*)**

Allocate a Accept-Encoding element.

**Parameters:**

*header* The element to work on.

**4.8.2.6 int accept\_encoding\_parse (accept\_encoding\_t \* *header*, char \* *hvalue*)**

Parse a Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.7 void accept\_encoding\_setelement (accept\_encoding\_t \* *header*, char \* *value*)**

Set the value of an Accept-Encoding element.

**Parameters:**

*header* The element to work on.

*value* The token value to set.

**4.8.2.8 int authorization\_2char (authorization\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Authorization element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.9 int authorization\_clone (authorization\_t \* *header*, authorization\_t \*\* *dest*)**

Clone a Authorization element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.10 void authorization\_free (authorization\_t \* *header*)**

Free a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.11 char\* authorization\_getalgorithm (authorization\_t \* *header*)**

Get value of the algorithm parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.12 char\* authorization\_getauth\_type (authorization\_t \* *header*)**

Get value of the auth\_type parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.13 char\* authorization\_getcnonce (authorization\_t \* *header*)**

Get value of the cnonce parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.14 char\* authorization\_getdigest (authorization\_t \* *header*)**

Get value of the digest parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.15 char\* authorization\_getmessage\_qop (authorization\_t \* *header*)**

Get value of the message\_qop parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.16 char\* authorization\_getnonce (authorization\_t \* header)**

Get value of the nonce parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.17 char\* authorization\_getnonce\_count (authorization\_t \* header)**

Get value of the nonce\_count parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.18 char\* authorization\_getopaque (authorization\_t \* header)**

Get value of the opaque parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.19 char\* authorization\_getrealm (authorization\_t \* header)**

Get value of the realm parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.20 char\* authorization\_getresponse (authorization\_t \* header)**

Get value of the response parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.21 char\* authorization\_geturi (authorization\_t \* header)**

Get value of the uri parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.22 char\* authorization\_getusername (authorization\_t \* header)**

Get value of the username parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.23 int authorization\_init (authorization\_t \*\* *header*)**

Allocate a Authorization element.

**Parameters:**

*header* The element to work on.

**4.8.2.24 int authorization\_parse (authorization\_t \* *header*, char \* *hvalue*)**

Parse a Authorization element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.25 void authorization\_setalgorithm (authorization\_t \* *header*, char \* *value*)**

Add the algorithm parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.26 void authorization\_setauth\_type (authorization\_t \* *header*, char \* *value*)**

Add the auth\_type parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.27 void authorization\_setcnonce (authorization\_t \* *header*, char \* *value*)**

Add the cnonce parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.28 void authorization\_setdigest (authorization\_t \* *header*, char \* *value*)**

Add the digest parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.



**4.8.2.29 void authorization\_setmessage\_qop (authorization\_t \* *header*, char \* *value*)**

Add the message\_qop parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.30 void authorization\_setnonce (authorization\_t \* *header*, char \* *value*)**

Add the nonce parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.31 void authorization\_setnonce\_count (authorization\_t \* *header*, char \* *value*)**

Add the nonce\_count parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.32 void authorization\_setopaque (authorization\_t \* *header*, char \* *value*)**

Add the opaque parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.33 void authorization\_setrealm (authorization\_t \* *header*, char \* *value*)**

Add the realm parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.34 void authorization\_setresponse (authorization\_t \* *header*, char \* *value*)**

Add the response parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.35 void authorization\_seturi (authorization\_t \* *header*, char \* *value*)**

Add the uri parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.36 void authorization\_setusername (authorization\_t \* *header*, char \* *value*)**

Add the username parameter from a Authorization element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.37 int body\_2char (body\_t \* *body*, char \*\* *dest*)**

Get a string representation of a body\_t element.

**Parameters:**

*body* The element to work on.

*dest* The resulting buffer.

**4.8.2.38 void body\_free (body\_t \* *body*)**

Free a body\_t element.

**Parameters:**

*body* The element to work on.

**4.8.2.39 int body\_init (body\_t \*\* *body*)**

Allocate a body\_t element.

**Parameters:**

*body* The element to work on.

**4.8.2.40 int body\_parse (body\_t \* *body*, char \* *buf*)**

Parse a body\_t element.

**Parameters:**

*body* The element to work on.

*buf* The buffer to parse.

**4.8.2.41 int body\_parse\_mime (body\_t \* *body*, char \* *buf*)**

Parse a body\_t element. (for mime message format) (NOT TESTED, use with care)

**Parameters:**

*body* The element to work on.

*buf* The buffer to parse.

**4.8.2.42 int call\_id\_2char (call\_id\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Call-id element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.43 int call\_id\_clone (call\_id\_t \* *header*, call\_id\_t \*\* *dest*)**

Clone a Call-id element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.44 void call\_id\_free (call\_id\_t \* *header*)**

Free a Call-id element.

**Parameters:**

*header* The element to work on.

**4.8.2.45 char\* call\_id\_gethost (call\_id\_t \* *header*)**

Get the host from a Call-Id header.

**Parameters:**

*header* The element to work on.

**4.8.2.46 char\* call\_id\_getnumber (call\_id\_t \* *header*)**

Get the number from a Call-Id header.

**Parameters:**

*header* The element to work on.

**4.8.2.47 int call\_id\_init (call\_id\_t \*\* *header*)**

Allocate a Call-id element.

**Parameters:**

*header* The element to work on.

**4.8.2.48 int call\_id\_parse (call\_id\_t \* *header*, char \* *hvalue*)**

Parse a Call-id element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.49 void call\_id\_sethost (call\_id\_t \* *header*, char \* *value*)**

Set the host in the Call-Id element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.50 void call\_id\_setnumber (call\_id\_t \* *header*, char \* *value*)**

Set the number in the Call-Id element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.51 int call\_info\_2char (call\_info\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Call-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.52 int call\_info\_clone (call\_info\_t \* *header*, call\_info\_t \*\* *dest*)**

Clone a Call-Info element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.53 void call\_info\_free (call\_info\_t \* *header*)**

Free a Call-Info element.

**Parameters:**

*header* The element to work on.

**4.8.2.54 char\* call\_info\_geturi (call\_info\_t \* *header*)**

Get the uri from a Call-Info header.

**Parameters:**

*header* The element to work on.

**4.8.2.55 int call\_info\_init (call\_info\_t \*\* *header*)**

Allocate a Call-Info element.

**Parameters:**

*header* The element to work on.

**4.8.2.56 int call\_info\_parse (call\_info\_t \* *header*, char \* *hvalue*)**

Parse a Call-Info element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.57 void call\_info\_seturi (call\_info\_t \* *header*, char \* *uri*)**

Set the uri in the Call-Info element.

**Parameters:**

*header* The element to work on.

*uri* The value of the element.

**4.8.2.58 int contact\_2char (contact\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Contact element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.59** `int contact_clone (contact_t * header, contact_t ** dest)`

Clone a Contact element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.60** `void contact_free (contact_t * header)`

Free a Contact element.

**Parameters:**

*header* The element to work on.

**4.8.2.61** `int contact_init (contact_t ** header)`

Allocate a Contact element.

**Parameters:**

*header* The element to work on.

**4.8.2.62** `int contact_parse (contact_t * header, char * hvalue)`

Parse a Contact element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.63** `int content_disposition_parse (content_disposition_t * header, char * hvalue)`

Parse a Content-Disposition element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.64** `int content_length_2char (content_length_t * header, char ** dest)`

Get a string representation of a Content-Length element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.65** `int content_length_clone (content_length_t * header, content_length_t ** dest)`

Clone a Content-Length element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.66** `void content_length_free (content_length_t * header)`

Free a Content-Length element.

**Parameters:**

*header* The element to work on.

**4.8.2.67** `int content_length_init (content_length_t ** header)`

Allocate a Content-Length element.

**Parameters:**

*header* The element to work on.

**4.8.2.68** `int content_length_parse (content_length_t * header, char * hvalue)`

Parse a Content-Length element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.69** `int content_type_2char (content_type_t * header, char ** dest)`

Get a string representation of a Content-Type element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.70** `int content_type_clone (content_type_t * header, content_type_t ** dest)`

Clone a Content-Type element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.71 void content\_type\_free (content\_type\_t \* *header*)**

Free a Content-Type element.

**Parameters:**

*header* The element to work on.

**4.8.2.72 int content\_type\_init (content\_type\_t \*\* *header*)**

Allocate a Content-Type element.

**Parameters:**

*header* The element to work on.

**4.8.2.73 int content\_type\_parse (content\_type\_t \* *header*, char \* *hvalue*)**

Parse a Content-Type element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.74 int cseq\_2char (cseq\_t \* *header*, char \*\* *dest*)**

Get a string representation of a CSeq element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.75 int cseq\_clone (cseq\_t \* *header*, cseq\_t \*\* *dest*)**

Clone a CSeq element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.76 void cseq\_free (cseq\_t \* *header*)**

Free a CSeq element.

**Parameters:**

*header* The element to work on.



**4.8.2.77 char\* cseq\_getmethod (cseq\_t \* *header*)**

Get the method from a CSeq header.

**Parameters:**

*header* The element to work on.

**4.8.2.78 char\* cseq\_getnumber (cseq\_t \* *header*)**

Get the number from a CSeq header.

**Parameters:**

*header* The element to work on.

**4.8.2.79 int cseq\_init (cseq\_t \*\* *header*)**

Allocate a CSeq element.

**Parameters:**

*header* The element to work on.

**4.8.2.80 int cseq\_parse (cseq\_t \* *header*, char \* *hvalue*)**

Parse a CSeq element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.81 void cseq\_setmethod (cseq\_t \* *header*, char \* *value*)**

Set the method in the CSeq element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.82 void cseq\_setnumber (cseq\_t \* *header*, char \* *value*)**

Set the number in the CSeq element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.83 int from\_2char (from\_t \* *header*, char \*\* *dest*)**

Get a string representation of a From element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.84 int from\_clone (from\_t \* *header*, from\_t \*\* *dest*)**

Clone a From element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.85 void from\_free (from\_t \* *header*)**

Free a From element.

**Parameters:**

*header* The element to work on.

**4.8.2.86 char\* from\_getdisplayname (from\_t \* *header*)**

Get the displayname from a From header.

**Parameters:**

*header* The element to work on.

**4.8.2.87 url\_t\* from\_geturl (from\_t \* *header*)**

Get the url from a From header.

**Parameters:**

*header* The element to work on.

**4.8.2.88 int from\_init (from\_t \*\* *header*)**

Allocate a From element.

**Parameters:**

*header* The element to work on.

**4.8.2.89 int from\_param\_get (from\_t \* *header*, int *pos*, generic\_param\_t \*\* *dest*)**

Get a header parameter from a From element.

**Parameters:**

- header* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the element found.

**4.8.2.90 int from\_parse (from\_t \* *header*, char \* *hvalue*)**

Parse a From element.

**Parameters:**

- header* The element to work on.
- hvalue* The string to parse.

**4.8.2.91 void from\_setdisplayname (from\_t \* *header*, char \* *value*)**

Set the displayname in the From element.

**Parameters:**

- header* The element to work on.
- value* The value of the element.

**4.8.2.92 void from\_seturl (from\_t \* *header*, url\_t \* *url*)**

Set the url in the From element.

**Parameters:**

- header* The element to work on.
- url* The value of the element.

**4.8.2.93 char\* generic\_param\_getname (generic\_param\_t \* *generic\_param*)**

Get the name of a generic parameter element.

**Parameters:**

- generic\_param* The element to work on.

**4.8.2.94 char\* generic\_param\_getvalue (generic\_param\_t \* *generic\_param*)**

Get the value of a generic parameter element.

**Parameters:**

- generic\_param* The element to work on.

**4.8.2.95** void generic\_param\_setname (generic\_param\_t \* *generic\_param*, char \* *name*)

Set the name of a generic parameter element.

**Parameters:**

*generic\_param* The element to work on.

*name* the token name to set.

**4.8.2.96** void generic\_param\_setvalue (generic\_param\_t \* *generic\_param*, char \* *value*)

Set the value of a generic parameter element.

**Parameters:**

*generic\_param* The element to work on.

*value* the token name to set.

**4.8.2.97** int header\_2char (header\_t \* *header*, char \*\* *dest*)

Get a string representation of a header element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated buffer.

**4.8.2.98** int header\_clone (header\_t \* *header*, header\_t \*\* *dest*)

Clone a header element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.99** void header\_free (header\_t \* *header*)

Free a header element.

**Parameters:**

*header* The element to work on.

**4.8.2.100** char\* header\_getname (header\_t \* *header*)

Get the token name a header element.

**Parameters:**

*header* The element to work on.

**4.8.2.101 char\* header\_getvalue (header\_t \* *header*)**

Get the token value a header element.

**Parameters:**

*header* The element to work on.

**4.8.2.102 int header\_init (header\_t \*\* *header*)**

Allocate a header element.

**Parameters:**

*header* The element to work on.

**4.8.2.103 void header\_setname (header\_t \* *header*, char \* *pname*)**

Set the token name a header element.

**Parameters:**

*header* The element to work on.

*pname* The token name to set.

**4.8.2.104 void header\_setvalue (header\_t \* *header*, char \* *pvalue*)**

Set the token value a header element.

**Parameters:**

*header* The element to work on.

*pvalue* The token value to set.

**4.8.2.105 int msg\_2char (sip\_t \* *sip*, char \*\* *dest*)**

Get a string representation of a sip\_t element.

**Parameters:**

*sip* The element to work on.

*dest* new allocated buffer returned.

**4.8.2.106 int msg\_clone (sip\_t \* *sip*, sip\_t \*\* *dest*)**

Clone a sip\_t element.

**Parameters:**

*sip* The element to clone.

*dest* The new allocated element cloned.

**4.8.2.107 int msg\_force\_update (sip\_t \* *sip*)**

Force a sip\_t element to be rebuild on next **msg\_2char()** (p. 134) call.

**Parameters:**

*sip* The element to work on.

**4.8.2.108 void msg\_free (sip\_t \* *sip*)**

Free all resource in a sip\_t element.

**Parameters:**

*sip* The element to free.

**4.8.2.109 int msg\_getaccept (sip\_t \* *sip*, int *pos*, accept\_t \*\* *dest*)**

Get one Accept header.

**Parameters:**

*sip* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the header found.

**4.8.2.110 int msg\_getaccept\_encoding (sip\_t \* *sip*, int *pos*, accept\_encoding\_t \*\* *dest*)**

Get one Accept-encoding header.

**Parameters:**

*sip* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the header found.

**4.8.2.111 int msg\_getaccept\_language (sip\_t \* *sip*, int *pos*, accept\_language\_t \*\* *dest*)**

Get one Accept header.

**Parameters:**

*sip* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the header found.

**4.8.2.112** `int msg_getalert_info (sip_t * sip, int pos, alert_info_t ** dest)`

Get one Alert-info header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.113** `int msg_getallow (sip_t * sip, int pos, allow_t ** dest)`

Get one Allow header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.114** `authorization_t* msg_getauthorization (sip_t * sip)`

Get one Authorisation header.

**Parameters:**

- sip* The element to work on.

**4.8.2.115** `int msg_getbody (sip_t * sip, int pos, body_t ** dest)`

Get one body header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the body found.

**4.8.2.116** `call_id_t* msg_getcall_id (sip_t * sip)`

Get one Call-id header.

**Parameters:**

- sip* The element to work on.

**4.8.2.117** `int msg_getcall_info (sip_t * sip, int pos, call_info_t ** dest)`

Get one Call-info header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.118 int msg\_getcontact (sip\_t \* *sip*, int *pos*, contact\_t \*\* *dest*)**

Get one Contact header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.119 int msg\_getcontent\_disposition (sip\_t \* *sip*, int *pos*, content\_disposition\_t \*\* *dest*)**

Get one Content-disposition header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.120 int msg\_getcontent\_encoding (sip\_t \* *sip*, int *pos*, content\_encoding\_t \*\* *dest*)**

Get one Content-encoding header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.121 content\_length\_t\* msg\_getcontent\_length (sip\_t \* *sip*)**

Get one Content-length header.

**Parameters:**

- sip* The element to work on.

**4.8.2.122 content\_type\_t\* msg\_getcontent\_type (sip\_t \* *sip*)**

Get one Content-type header.

**Parameters:**

- sip* The element to work on.

**4.8.2.123 cseq\_t\* msg\_getcseq (sip\_t \* *sip*)**

Get one Cseq header.

**Parameters:**

- sip* The element to work on.



**4.8.2.124 int msg\_geterror\_info (sip\_t \* sip, int pos, error\_info\_t \*\* dest)**

Get one Error-info header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.125 from\_t\* msg\_getfrom (sip\_t \* sip)**

Get the From header.

**Parameters:**

- sip* The element to work on.

**4.8.2.126 int msg\_getheader (sip\_t \* sip, int pos, header\_t \*\* dest)**

Get one "unknown" header.

**Parameters:**

- sip* The element to work on.
- pos* The index of the element to get.
- dest* A pointer on the header found.

**4.8.2.127 char\* msg\_getmethod (sip\_t \* sip)**

Get the method name.

**Parameters:**

- sip* The element to work on.

**4.8.2.128 mime\_version\_t\* msg\_getmime\_version (sip\_t \* sip)**

Get the Mime-version header.

**Parameters:**

- sip* The element to work on.

**4.8.2.129 proxy\_authenticate\_t\* msg\_getproxy\_authenticate (sip\_t \* sip)**

Get the Proxy-authenticate header.

**Parameters:**

- sip* The element to work on.

**4.8.2.130** `int msg_getproxy_authorization (sip_t * sip, int pos, proxy_authorization_t ** dest)`

Get one Proxy-authorization header.

**Parameters:**

*sip* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the header found.

**4.8.2.131** `char* msg_getreason (int status_code)`

Get the usual reason phrase as defined in SIP for a specific status code.

**Parameters:**

*status\_code* A status code.

**4.8.2.132** `char* msg_getreasonphrase (sip_t * sip)`

Get the reason phrase. This is entirely free in SIP.

**Parameters:**

*sip* The element to work on.

**4.8.2.133** `int msg_getrecord_route (sip_t * sip, int pos, record_route_t ** dest)`

Get one Record-route header.

**Parameters:**

*sip* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the header found.

**4.8.2.134** `int msg_getroute (sip_t * sip, int pos, route_t ** dest)`

Get one Route header.

**Parameters:**

*sip* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the header found.

**4.8.2.135** `char* msg_getstatuscode (sip_t * sip)`

Get the status code.

**Parameters:**

*sip* The element to work on.

**4.8.2.136 to\_t\* msg\_getto (sip\_t \* sip)**

Get the To header.

**Parameters:**

*sip* The element to work on.

**4.8.2.137 url\_t\* msg\_geturi (sip\_t \* sip)**

Get the Request-URI.

**Parameters:**

*sip* The element to work on.

**4.8.2.138 char\* msg\_getversion (sip\_t \* sip)**

Get the SIP version.

**Parameters:**

*sip* The element to work on.

**4.8.2.139 int msg\_getvia (sip\_t \* sip, int pos, via\_t \*\* dest)**

Get one Via header.

**Parameters:**

*sip* The element to work on.

*pos* The index of the element to get.

*dest* A pointer on the header found.

**4.8.2.140 www\_authenticate\_t\* msg\_getwww\_authenticate (sip\_t \* sip)**

Get one Www-authenticate header.

**Parameters:**

*sip* The element to work on.

**4.8.2.141 int msg\_header\_getbyname (sip\_t \* sip, char \* hname, int pos, header\_t \*\* dest)**

Find an "unknown" header. (not defined in oSIP)

**Parameters:**

*sip* The element to work on.

*hname* The name of the header to find.

*pos* The index where to start searching for the header.

*dest* A pointer to the header found.

**4.8.2.142** `int msg_init (sip_t ** sip)`

Allocate a sip\_t element.

**Parameters:**

*sip* The element to allocate.

**4.8.2.143** `int msg_parse (sip_t * sip, char * message)`

Parse a sip\_t element.

**Parameters:**

*sip* The resulting element.

*message* The buffer to parse.

**4.8.2.144** `int msg_setaccept (sip_t * sip, char * hvalue)`

Set the Accept header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.145** `int msg_setaccept_encoding (sip_t * sip, char * hvalue)`

Set the Accept-encoding header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.146** `int msg_setaccept_language (sip_t * sip, char * hvalue)`

Set the Accept-language header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.147** `int msg_setalert_info (sip_t * sip, char * hvalue)`

Set the Alert-info header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.148** `int msg_setallow (sip_t * sip, char * hvalue)`

Set the Allow header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.149** `int msg_setauthorization (sip_t * sip, char * hvalue)`

Set the Authorisation header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.150** `int msg_setbody (sip_t * sip, char * buf)`

Set the Body of the SIP message.

**Parameters:**

- sip* The element to work on.
- buf* The string containing the body.

**4.8.2.151** `int msg_setbody_mime (sip_t * sip, char * buf)`

Set a type for a body. (NOT TESTED! use with care)

**Parameters:**

- sip* The element to work on.
- buf* the mime type of body.

**4.8.2.152** `int msg_setcall_id (sip_t * sip, char * hvalue)`

Set the Call-id header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.153** `int msg_setcall_info (sip_t * sip, char * hvalue)`

Set the Call-info header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.154 int msg\_setcontact (sip\_t \* *sip*, char \* *hvalue*)**

Set the Contact header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.155 int msg\_setcontent\_disposition (sip\_t \* *sip*, char \* *hvalue*)**

Set the Content-disposition header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.156 int msg\_setcontent\_encoding (sip\_t \* *sip*, char \* *hvalue*)**

Set the Content-encoding header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.157 int msg\_setcontent\_length (sip\_t \* *sip*, char \* *hvalue*)**

Set the Content-length header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.158 int msg\_setcontent\_type (sip\_t \* *sip*, char \* *hvalue*)**

Set the Content-type header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.159 int msg\_setcseq (sip\_t \* *sip*, char \* *hvalue*)**

Set the Cseq header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.160 int msg\_seterror\_info (sip\_t \* *sip*, char \* *hvalue*)**

Set the Error-info header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.161 int msg\_setfrom (sip\_t \* *sip*, char \* *hvalue*)**

Set the From header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.162 int msg\_setheader (sip\_t \* *sip*, char \* *hname*, char \* *hvalue*)**

Allocate and Add an "unknown" header (not defined in oSIP).

**Parameters:**

- sip* The element to work on.
- hname* The token name.
- hvalue* The token value.

**4.8.2.163 void msg\_setmethod (sip\_t \* *sip*, char \* *method*)**

Set the method. You can set any string here.

**Parameters:**

- sip* The element to work on.
- method* The method name.

**4.8.2.164 int msg\_setmime\_version (sip\_t \* *sip*, char \* *hvalue*)**

Set the mime-version header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.165 int msg\_setproxy\_authenticate (sip\_t \* *sip*, char \* *hvalue*)**

Set the Proxy-authenticate header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.166** `int msg_setproxy_authorization (sip_t * sip, char * hvalue)`

Set the Proxy-authorization header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.167** `void msg_setreasonphrase (sip_t * sip, char * reason)`

Set the reason phrase. This is entirely free in SIP.

**Parameters:**

- sip* The element to work on.
- reason* The reason phrase.

**4.8.2.168** `int msg_setrecord_route (sip_t * sip, char * hvalue)`

Set the Record-Route header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.169** `int msg_setroute (sip_t * sip, char * hvalue)`

Set the Route header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.

**4.8.2.170** `void msg_setstatuscode (sip_t * sip, char * statuscode)`

Set the status code. This is entirely free in SIP.

**Parameters:**

- sip* The element to work on.
- statuscode* The status code.

**4.8.2.171** `int msg_setto (sip_t * sip, char * hvalue)`

Set the To header.

**Parameters:**

- sip* The element to work on.
- hvalue* The string describing the element.



**4.8.2.172 void msg\_seturi (sip\_t \* *sip*, url\_t \* *uri*)**

Set the Request-URI.

**Parameters:**

*sip* The element to work on.

*uri* The uri to set.

**4.8.2.173 void msg\_setversion (sip\_t \* *sip*, char \* *version*)**

Set the SIP version used. (use "SIP/2.0")

**Parameters:**

*sip* The element to work on.

*version* The version of SIP.

**4.8.2.174 int msg\_setvia (sip\_t \* *sip*, char \* *hvalue*)**

Set the Via header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.175 int msg\_setwww\_authenticate (sip\_t \* *sip*, char \* *hvalue*)**

Set the Www-authenticate header.

**Parameters:**

*sip* The element to work on.

*hvalue* The string describing the element.

**4.8.2.176 int parser\_init ()**

Initialise the oSIP parser.

**4.8.2.177 int record\_route\_2char (record\_route\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Record-Route element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.178 void record\_route\_free (record\_route\_t \* *header*)**

Free a Record-Route element.

**Parameters:**

*header* The element to work on.

**4.8.2.179 int record\_route\_init (record\_route\_t \*\* *header*)**

Allocate a Record-Route element.

**Parameters:**

*header* The element to work on.

**4.8.2.180 int record\_route\_parse (record\_route\_t \* *header*, char \* *hvalue*)**

Parse a Record-Route element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.181 int route\_2char (route\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Route element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.182 void route\_free (route\_t \* *header*)**

Free a Route element.

**Parameters:**

*header* The element to work on.

**4.8.2.183 int route\_init (route\_t \*\* *header*)**

Allocate a Route element.

**Parameters:**

*header* The element to work on.

**4.8.2.184 int route\_parse (route\_t \* header, char \* hvalue)**

Parse a Route element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.185 int to\_2char (to\_t \* header, char \*\* dest)**

Get a string representation of a To element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.186 int to\_clone (to\_t \* header, to\_t \*\* dest)**

Clone a To element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.187 void to\_free (to\_t \* header)**

Free a To element.

**Parameters:**

*header* The element to work on.

**4.8.2.188 int to\_init (to\_t \*\* header)**

Allocate a To element.

**Parameters:**

*header* The element to work on.

**4.8.2.189 int to\_parse (to\_t \* header, char \* hvalue)**

Parse a To element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.190** `int via_2char (via_t * header, char ** dest)`

Get a string representation of a Via element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.191** `int via_clone (via_t * header, via_t ** dest)`

Clone a Via element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.192** `void via_free (via_t * header)`

Free a Via element.

**Parameters:**

*header* The element to work on.

**4.8.2.193** `char* via_getcomment (via_t * header)`

Get the comment from a Via header.

**Parameters:**

*header* The element to work on.

**4.8.2.194** `char* via_gethost (via_t * header)`

Get the host from a Via header.

**Parameters:**

*header* The element to work on.

**4.8.2.195** `char* via_getport (via_t * header)`

Get the port from a Via header.

**Parameters:**

*header* The element to work on.

**4.8.2.196** `char* via_getprotocol (via_t * header)`

Get the protocol from a Via header.

**Parameters:**

*header* The element to work on.

**4.8.2.197 char\* via\_getversion (via\_t \* header)**

Get the SIP version from a Via header.

**Parameters:**

*header* The element to work on.

**4.8.2.198 int via\_init (via\_t \*\* header)**

Allocate a Via element.

**Parameters:**

*header* The element to work on.

**4.8.2.199 int via\_parse (via\_t \* header, char \* hvalue)**

Parse a Via element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.200 void via\_setcomment (via\_t \* header, char \* value)**

Set the comment in the Via element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.201 void via\_sethost (via\_t \* header, char \* value)**

Set the host in the Via element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.202 void via\_setport (via\_t \* header, char \* value)**

Set the port in the Via element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.203 void via\_setprotocol (via\_t \* *header*, char \* *value*)**

Set the protocol in the Via element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.204 void via\_setversion (via\_t \* *header*, char \* *value*)**

Set the SIP version in the Via element.

**Parameters:**

*header* The element to work on.

*value* The value of the element.

**4.8.2.205 int www\_authenticate\_2char (www\_authenticate\_t \* *header*, char \*\* *dest*)**

Get a string representation of a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the new allocated string.

**4.8.2.206 int www\_authenticate\_clone (www\_authenticate\_t \* *header*,  
www\_authenticate\_t \*\* *dest*)**

Clone a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*dest* A pointer on the copy of the element.

**4.8.2.207 void www\_authenticate\_free (www\_authenticate\_t \* *header*)**

Free a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.208 char\* www\_authenticate\_getalgorithm (www\_authenticate\_t \* *header*)**

Get value of the algorithm parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.209 char\* www\_authenticate\_getauth\_type (www\_authenticate\_t \* header)**

Get value of the auth\_type parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.210 char\* www\_authenticate\_getdomain (www\_authenticate\_t \* header)**

Get value of the domain parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.211 char\* www\_authenticate\_getnonce (www\_authenticate\_t \* header)**

Get value of the nonce parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.212 char\* www\_authenticate\_getopaque (www\_authenticate\_t \* header)**

Get value of the opaque parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.213 char\* www\_authenticate\_getqop\_options (www\_authenticate\_t \* header)**

Get value of the qop\_options parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.214 char\* www\_authenticate\_getrealm (www\_authenticate\_t \* header)**

Get value of the realm parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.215 char\* www\_authenticate\_getstale (www\_authenticate\_t \* header)**

Get value of the stale parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.216** `int www_authenticate_init (www_authenticate_t ** header)`

Allocate a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

**4.8.2.217** `int www_authenticate_parse (www_authenticate_t * header, char * hvalue)`

Parse a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*hvalue* The string to parse.

**4.8.2.218** `void www_authenticate_setalgorithm (www_authenticate_t * header, char * value)`

Add the algorithm parameter in a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.219** `void www_authenticate_setauth_type (www_authenticate_t * header, char * value)`

Add the auth\_type parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.220** `void www_authenticate_setdomain (www_authenticate_t * header, char * value)`

Add the domain parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.221** `void www_authenticate_setnonce (www_authenticate_t * header, char * value)`

Add the nonce parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.



**4.8.2.222** void `www_authenticate_setopaque` (`www_authenticate_t` \* *header*, char \* *value*)

Add the opaque parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.223** void `www_authenticate_setqop_options` (`www_authenticate_t` \* *header*, char \* *value*)

Add the qop\_options parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.224** void `www_authenticate_setrealm` (`www_authenticate_t` \* *header*, char \* *value*)

Add the realm parameter from a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

**4.8.2.225** void `www_authenticate_setstale` (`www_authenticate_t` \* *header*, char \* *value*)

Add the stale parameter in a Www-Authenticate element.

**Parameters:**

*header* The element to work on.

*value* The value of the new parameter.

## 4.9 oSIP type definitions

### Compounds

- struct **accept\_encoding\_t**
- struct **authorization\_t**
- struct **body\_t**
- struct **call\_id\_t**
- struct **call\_info\_t**
- struct **content\_length\_t**
- struct **content\_type\_t**
- struct **cseq\_t**
- struct **from\_t**
- struct **header\_t**
- struct **language\_tag\_t**
- struct **sip\_t**
- struct **startline\_t**
- struct **via\_t**
- struct **www\_authenticate\_t**

### Defines

- **#define BODY\_MESSAGE\_MAX\_SIZE 500**

### Typedefs

- typedef startline\_t **startline\_t**
- typedef header\_t **header\_t**
- typedef cseq\_t **cseq\_t**
- typedef via\_t **via\_t**
- typedef url\_param\_t **generic\_param\_t**
- typedef from\_t **from\_t**
- typedef **from\_t** **to\_t**
- typedef **from\_t** **contact\_t**
- typedef **from\_t** **record\_route\_t**
- typedef **from\_t** **route\_t**
- typedef call\_id\_t **call\_id\_t**
- typedef content\_length\_t **content\_length\_t**
- typedef language\_tag\_t **language\_tag\_t**
- typedef **content\_length\_t** **allow\_t**
- typedef **content\_length\_t** **content\_encoding\_t**
- typedef **content\_length\_t** **mime\_version\_t**
- typedef content\_type\_t **content\_type\_t**
- typedef **content\_type\_t** **accept\_t**
- typedef accept\_encoding\_t **accept\_encoding\_t**
- typedef **accept\_encoding\_t** **accept\_language\_t**
- typedef call\_info\_t **call\_info\_t**
- typedef **call\_info\_t** **alert\_info\_t**
- typedef **call\_info\_t** **error\_info\_t**

- typedef **call\_info\_t** **content\_disposition\_t**
- typedef **call\_info\_t** **encryption\_t**
- typedef **www\_authenticate\_t** **www\_authenticate\_t**
- typedef **www\_authenticate\_t** **proxy\_authenticate\_t**
- typedef **authorization\_t** **authorization\_t**
- typedef **authorization\_t** **proxy\_authorization\_t**
- typedef **body\_t** **body\_t**
- typedef **sip\_t** **sip\_t**

#### 4.9.1 Define Documentation

##### 4.9.1.1 #define **BODY\_MESSAGE\_MAX\_SIZE** 500

You can define the maximum authorised length for a body inside a SIP message.

#### 4.9.2 Typedef Documentation

##### 4.9.2.1 typedef struct **accept\_encoding\_t** **accept\_encoding\_t**

Structure for Accept-Encoding headers. @defvar **accept\_encoding\_t**

##### 4.9.2.2 typedef **accept\_encoding\_t** **accept\_language\_t**

Structure for Accept-Language headers. @defvar **accept\_language\_t**

##### 4.9.2.3 typedef **content\_type\_t** **accept\_t**

Structure for accept headers. @defvar **accept\_t**

##### 4.9.2.4 typedef **call\_info\_t** **alert\_info\_t**

Structure for Alert-Info headers. @defvar **alert\_info\_t**

##### 4.9.2.5 typedef **content\_length\_t** **allow\_t**

Structure for Allow headers. @defvar **allow\_t**

##### 4.9.2.6 typedef struct **authorization\_t** **authorization\_t**

Structure for Authorization headers. @defvar **authorization\_t**

##### 4.9.2.7 typedef struct **body\_t** **body\_t**

Structure for Body - LIGHT SUPPORT FOR MIME FORMAT: TO BE TESTED-. @defvar **body\_t**

**4.9.2.8    typedef struct call\_id\_t call\_id\_t**

Structure for Call-Id headers. @defvar call\_id\_t

**4.9.2.9    typedef struct call\_info\_t call\_info\_t**

Structure for Call-Info headers. @defvar call\_info\_t

**4.9.2.10    typedef struct from\_t contact\_t**

Structure for Contact headers. @defvar contact\_t

**4.9.2.11    typedef struct call\_info\_t content\_disposition\_t**

Structure for Content-Disposition headers. @defvar content\_disposition\_t

**4.9.2.12    typedef struct content\_length\_t content\_encoding\_t**

Structure for Content-Encoding headers. @defvar content\_encoding\_t

**4.9.2.13    typedef struct content\_length\_t content\_length\_t**

Structure for Content-Length headers. @defvar content\_length\_t

**4.9.2.14    typedef struct content\_type\_t content\_type\_t**

Structure for Content-Type headers. @defvar content\_type\_t

**4.9.2.15    typedef struct cseq\_t cseq\_t**

Structure for CSeq headers. @defvar cseq\_t

**4.9.2.16    typedef struct call\_info\_t encryption\_t**

Structure for encryption headers. - NOT IMPLEMENTED - @defvar encryption\_t

**4.9.2.17    typedef struct call\_info\_t error\_info\_t**

Structure for Error-Info headers. @defvar error\_info\_t

**4.9.2.18    typedef struct from\_t from\_t**

Structure for From headers. @defvar from\_t

**4.9.2.19    typedef url\_param\_t generic\_param\_t**

Structure for generic parameter headers. Generic parameter are used in a lot of headers. (To, From, Route, Record-Route...) All those headers use a common API but this is hidden by MACROs that you can be found in **smsg.h**. @defvar cseq\_t

**4.9.2.20    typedef struct header\_t header\_t**

Structure for 'unknown' headers. NOTE: 'unknown' header' are used in oSIP for all header that are not defined by oSIP in the sip\_t structure. This means that all 'unknown' header has to be handled with the API related to this structure. @defvar startline\_t

**4.9.2.21    typedef struct language\_tag\_t language\_tag\_t**

Structure for Language-Tag headers. - NOT IMPLEMENTED - @defvar language\_tag\_t

**4.9.2.22    typedef content\_length\_t mime\_version\_t**

Structure for Mime-Version headers. @defvar mime\_version\_t

**4.9.2.23    typedef www\_authenticate\_t proxy\_authenticate\_t**

Structure for Proxy-Authenticate headers. @defvar proxy\_authenticate\_t

**4.9.2.24    typedef authorization\_t proxy\_authorization\_t**

Structure for Proxy-Authorization headers. @defvar proxy\_authorization\_t

**4.9.2.25    typedef from\_t record\_route\_t**

Structure for Record-Route headers. @defvar record\_route\_t

**4.9.2.26    typedef from\_t route\_t**

Structure for Route headers. @defvar route\_t

**4.9.2.27    typedef struct sip\_t sip\_t**

Structure for SIP Message (REQUEST and RESPONSE). @defvar sip\_t

**4.9.2.28    typedef struct startline\_t startline\_t**

Structure for startline (1st line of SIP message either REQUEST and RESPONSE). @defvar startline\_t

**4.9.2.29    typedef from\_t to\_t**

Structure for To headers. @defvar to\_t

**4.9.2.30 typedef struct via\_t via\_t**

Structure for Via headers. @defvar via\_t

**4.9.2.31 typedef struct www\_authenticate\_t www\_authenticate\_t**

Structure for WWW-Authenticate headers. @defvar www\_authenticate\_t

## 4.10 oSIP Thread Routines

### Typedefs

- `typedef pthread_t sthread_t`

### Functions

- `sthread_t * sthread_create` (int *stacksize*, `sthread_t *thread`, void `*(func)(void *)`, void `*arg`)
- `int sthread_join` (`sthread_t *thread`)
- `int sthread_setpriority` (`sthread_t *thread`, int *priority*)
- `void sthread_exit` ()

#### 4.10.1 Typedef Documentation

##### 4.10.1.1 sthread\_t

Structure for referencing a thread

#### 4.10.2 Function Documentation

##### 4.10.2.1 sthread\_t\* sthread\_create (int *stacksize*, sthread\_t \* *thread*, void `*(func)(void *)`, void \* *arg*)

Allocate (or initialise if a thread address is given)

###### Parameters:

*stacksize* The stack size of the thread. (20000 is a good value)

*thread* The thread to create. (if it is NULL, a new thread is returned)

*func* The method where the thread start.

*arg* A pointer on the argument given to the method 'func'.

##### 4.10.2.2 void sthread\_exit ()

Exit from a thread.

##### 4.10.2.3 int sthread\_join (sthread\_t \* *thread*)

Join a thread.

###### Parameters:

*thread* The thread to join.

#### 4.10.2.4 int sthread\_setpriority (sthread\_t \* *thread*, int *priority*)

Set the priority of a thread.

**Parameters:**

*thread* The thread to work on.

*priority* The priority value to set.



## 4.11 oSIP url parser Handling

### Compounds

- struct **url\_param\_t**
- struct **url\_t**

### Defines

- **#define url\_header\_init**(url\_header) url\_param\_init(url\_header)
- **#define url\_header\_free**(url\_header) url\_param\_free(url\_header)
- **#define url\_header\_set**(url\_header, name, value) url\_param\_set(url\_header, name, value)
- **#define url\_header\_clone**(url\_header, dest) url\_param\_clone(url\_header, dest)
- **#define url\_header\_add**(url\_headers, name, value) url\_param\_add(url\_headers, name, value)
- **#define url\_header\_getbyname**(url\_headers, name, dest) url\_param\_getbyname(url\_headers, name, dest)
- **#define url\_set\_transport\_udp**(url) url\_param\_add(url → url\_params, "transport", "udp")
- **#define url\_set\_transport\_tcp**(url) url\_param\_add(url → url\_params, "transport", "tcp")
- **#define url\_set\_transport\_sctp**(url) url\_param\_add(url → url\_params, "transport", "sctp")
- **#define url\_set\_transport\_tls**(url) url\_param\_add(url → url\_params, "transport", "tls")
- **#define url\_set\_transport**(url, value) url\_param\_add(url → url\_params, "transport", value)
- **#define url\_set\_user\_phone**(url) url\_param\_add(url → url\_params, "user", "phone")
- **#define url\_set\_user\_ip**(url) url\_param\_add(url → url\_params, "user", "ip")
- **#define url\_set\_user**(url, value) url\_param\_add(url → url\_params, "user", value)
- **#define url\_set\_method\_invite**(url) url\_param\_add(url → url\_params, "method", "INVITE")
- **#define url\_set\_method\_ack**(url) url\_param\_add(url → url\_params, "method", "ACK")
- **#define url\_set\_method\_options**(url) url\_param\_add(url → url\_params, "method", "OPTIONS")
- **#define url\_set\_method\_bye**(url) url\_param\_add(url → url\_params, "method", "BYE")
- **#define url\_set\_method\_cancel**(url) url\_param\_add(url → url\_params, "method", "CANCEL")
- **#define url\_set\_method\_register**(url) url\_param\_add(url → url\_params, "method", "REGISTER")
- **#define url\_set\_method**(url, value) url\_param\_add(url → url\_params, "method", value)
- **#define url\_set\_ttl**(url, value) url\_param\_add(url → url\_params, "ttl", value)
- **#define url\_set\_maddr**(url, value) url\_param\_add(url → url\_params, "maddr", value)
- **#define url\_uparam\_get**(url, pos, dest) url\_param\_get(url → url\_params, pos, dest)
- **#define url\_uparam\_add**(url, name, value) url\_param\_add(url → url\_params, name, value)
- **#define url\_uparam\_getbyname**(url, name, dest) url\_param\_getbyname(url → url\_params, name, dest)
- **#define url\_uheader\_get**(url, pos, dest) url\_header\_get(url → url\_headers, pos, dest)
- **#define url\_uheader\_add**(url, name, value) url\_header\_add(url → url\_headers, name, value)
- **#define url\_uheader\_getbyname**(url, name, dest) url\_header\_getbyname(url → url\_headers, name, dest)

## Typedefs

- typedef url\_param\_t **url\_param\_t**
- typedef url\_param\_t **url\_header\_t**
- typedef url\_t **url\_t**

## Functions

- int **url\_param\_init** (url\_param\_t \*\*url\_param)
- void **url\_param\_free** (url\_param\_t \*url\_param)
- int **url\_param\_set** (url\_param\_t \*url\_param, char \*name, char \*value)
- int **url\_param\_clone** (url\_param\_t \*url\_param, url\_param\_t \*\*dest)
- int **url\_param\_add** (list\_t \*url\_params, char \*name, char \*value)
- int **url\_param\_getbyname** (list\_t \*url\_params, char \*name, url\_param\_t \*\*dest)
- int **url\_init** (url\_t \*\*url)
- void **url\_free** (url\_t \*url)
- int **url\_parse** (url\_t \*url, char \*buf)
- int **url\_2char** (url\_t \*url, char \*\*dest)
- int **url\_clone** (url\_t \*url, url\_t \*\*dest)
- void **url\_setscheme** (url\_t \*url, char \*value)
- char \* **url\_getscheme** (url\_t \*url)
- void **url\_sethost** (url\_t \*url, char \*value)
- char \* **url\_gethost** (url\_t \*url)
- void **url\_setusername** (url\_t \*url, char \*value)
- char \* **url\_getusername** (url\_t \*url)
- void **url\_setpassword** (url\_t \*url, char \*value)
- char \* **url\_getpassword** (url\_t \*url)
- void **url\_setport** (url\_t \*url, char \*value)
- char \* **url\_getport** (url\_t \*url)

### 4.11.1 Define Documentation

#### 4.11.1.1 #define url\_header\_add(url\_headers, name, value) url\_param\_add(url\_headers,name,value)

Allocate and add a generic parameter element in a list.

**Parameters:**

- url\_headers* The list of generic parameter element to work on.
- name* The token name.
- value* The token value.

#### 4.11.1.2 #define url\_header\_clone(url\_header, dest) url\_param\_clone(url\_header,dest)

Clone a generic parameter element.

**Parameters:**

- url\_header* The element to work on.
- dest* The resulting new allocated element.

**4.11.1.3** `#define url_header_free(url_header) url_param_free(url_header)`

Free a generic parameter element.

**Parameters:**

*url\_header* The element to work on.

**4.11.1.4** `#define url_header_getbyname(url_headers, name, dest)  
url_param_getbyname(url_headers,name,dest)`

Find in a generic parameter element in a list.

**Parameters:**

*url\_headers* The list of generic parameter element to work on.

*name* The name of the parameter element to find.

*dest* A pointer on the element found.

**4.11.1.5** `#define url_header_init(url_header) url_param_init(url_header)`

Allocate a generic parameter element.

**Parameters:**

*url\_header* The element to work on.

**4.11.1.6** `#define url_header_set(url_header, name, value) url_param_set(url_header,  
name, value)`

Set values of a generic parameter element.

**Parameters:**

*url\_header* The element to work on.

*name* The token name.

*value* The token value.

**4.11.1.7** `#define url_set_maddr(url, value) url_param_add(url → url_params,  
"maddr", value)`

Set a maddr parameter in a url element.

**Parameters:**

*url* The element to work on.

*value* The value for the maddr parameter.

**4.11.1.8** `#define url_set_method(url, value) url_param_add(url → url_params,  
"method", value)`

Set a method parameter in a url element.

**Parameters:**

*url* The element to work on.

*value* The value for the method parameter.

**4.11.1.9** `#define url_set_method_ack(url) url_param_add(url → url_params,  
"method", "ACK")`

Set a method parameter to ACK in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.10** `#define url_set_method_bye(url) url_param_add(url → url_params,  
"method", "BYE")`

Set a method parameter to BYE in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.11** `#define url_set_method_cancel(url) url_param_add(url → url_params,  
"method", "CANCEL")`

Set a method parameter to CANCEL in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.12** `#define url_set_method_invite(url) url_param_add(url → url_params,  
"method", "INVITE")`

Set a method parameter to INVITE in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.13** `#define url_set_method_options(url) url_param_add(url → url_params,  
"method", "OPTIONS")`

Set a method parameter to OPTIONS in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.14** `#define url_set_method_register(url) url_param_add(url → url_params, "method", "REGISTER")`

Set a method parameter to REGISTER in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.15** `#define url_set_transport(url, value) url_param_add(url → url_params, "transport", value)`

Set the transport parameter to TLS in a url element.

**Parameters:**

*url* The element to work on.

*value* The value describing the transport protocol.

**4.11.1.16** `#define url_set_transport_sctp(url) url_param_add(url → url_params, "transport", "sctp")`

Set the transport parameter to SCTP in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.17** `#define url_set_transport_tcp(url) url_param_add(url → url_params, "transport", "tcp")`

Set the transport parameter to TCP in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.18** `#define url_set_transport_tls(url) url_param_add(url → url_params, "transport", "tls")`

Set the transport parameter to TLS in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.19** `#define url_set_transport_udp(url) url_param_add(url → url_params, "transport", "udp")`

Set the transport parameter to UDP in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.20** `#define url_set_ttl(url, value) url_param_add(url → url_params, "ttl", value)`

Set a ttl parameter in a url element.

**Parameters:**

*url* The element to work on.

*value* The value for the ttl parameter.

**4.11.1.21** `#define url_set_user(url, value) url_param_add(url → url_params, "user", value)`

Set the user parameter in a url element.

**Parameters:**

*url* The element to work on.

*value* The value describing the user url.

**4.11.1.22** `#define url_set_user_ip(url) url_param_add(url → url_params, "user", "ip")`

Set the user parameter to IP in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.23** `#define url_set_user_phone(url) url_param_add(url → url_params, "user", "phone")`

Set the user parameter to PHONE in a url element.

**Parameters:**

*url* The element to work on.

**4.11.1.24** `#define url_uheader_add(url, name, value) url_header_add(url → url_headers, name, value)`

Allocate and add a url header element in a url element.

**Parameters:**

*url* The element to work on.

*name* The token name.

*value* The token value.

**4.11.1.25** `#define url_uheader_get(url, pos, dest) url_header_get(url → url_headers,pos,dest)`

Get a url header in a url element.

**Parameters:**

*url* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the header found.

**4.11.1.26** `#define url_uheader_getbyname(url, name, dest) url_header_getbyname(url → url_headers,name,dest)`

Find in a url header element in a url element.

**Parameters:**

*url* The element to work on.  
*name* The name of the url header element to find.  
*dest* A pointer on the element found.

**4.11.1.27** `#define url_uparam_add(url, name, value) url_param_add(url → url_params,name,value)`

Allocate and add a url parameter element in a url element.

**Parameters:**

*url* The element to work on.  
*name* The token name.  
*value* The token value.

**4.11.1.28** `#define url_uparam_get(url, pos, dest) url_param_get(url → url_params,pos,dest)`

Get a url parameter in a url element.

**Parameters:**

*url* The element to work on.  
*pos* The index of the element to get.  
*dest* A pointer on the header found.

**4.11.1.29** `#define url_uparam_getbyname(url, name, dest) url_param_getbyname(url → url_params,name,dest)`

Find in a url parameter element in a url element.

**Parameters:**

*url* The element to work on.  
*name* The name of the url parameter element to find.  
*dest* A pointer on the element found.

### 4.11.2 Typedef Documentation

#### 4.11.2.1 typedef url\_param\_t url\_header\_t

Structure for referencing url headers. @defvar url\_header\_t

#### 4.11.2.2 typedef struct url\_param\_t url\_param\_t

Structure for referencing url parameters. @defvar url\_param\_t

#### 4.11.2.3 typedef struct url\_t url\_t

Structure for referencing SIP urls. @defvar url\_t

### 4.11.3 Function Documentation

#### 4.11.3.1 int url\_2char (url\_t \* *url*, char \*\* *dest*)

Get a string representation of a url element.

**Parameters:**

*url* The element to work on.

*dest* The resulting new allocated buffer.

#### 4.11.3.2 int url\_clone (url\_t \* *url*, url\_t \*\* *dest*)

Clone a url element.

**Parameters:**

*url* The element to work on.

*dest* The resulting new allocated element.

#### 4.11.3.3 void url\_free (url\_t \* *url*)

Free a url element.

**Parameters:**

*url* The element to work on.

#### 4.11.3.4 char \* url\_gethost (url\_t \* *url*)

Get the host of a url element.

**Parameters:**

*url* The element to work on.



**4.11.3.5 char\* url\_getpassword (url\_t \* *url*)**

Get the password of a url element.

**Parameters:**

*url* The element to work on.

**4.11.3.6 char\* url\_getport (url\_t \* *url*)**

Get the port of a url element.

**Parameters:**

*url* The element to work on.

**4.11.3.7 char\* url\_getscheme (url\_t \* *url*)**

Get the scheme of a url element.

**Parameters:**

*url* The element to work on.

**4.11.3.8 char\* url\_getusername (url\_t \* *url*)**

Get the username of a url element.

**Parameters:**

*url* The element to work on.

**4.11.3.9 int url\_init (url\_t \*\* *url*)**

Allocate a url element.

**Parameters:**

*url* The element to work on.

**4.11.3.10 int url\_param\_add (list\_t \* *url\_params*, char \* *name*, char \* *value*)**

Allocate and add a url parameter element in a list.

**Parameters:**

*url\_params* The list of url parameter element to work on.

*name* The token name.

*value* The token value.

**4.11.3.11 int url\_param\_clone (url\_param\_t \* *url\_param*, url\_param\_t \*\* *dest*)**

Clone a url parameter element.

**Parameters:**

- url\_param* The element to work on.
- dest* The resulting new allocated element.

**4.11.3.12 void url\_param\_free (url\_param\_t \* *url\_param*)**

Free a url parameter element.

**Parameters:**

- url\_param* The element to work on.

**4.11.3.13 int url\_param\_getbyname (list\_t \* *url\_params*, char \* *name*, url\_param\_t \*\* *dest*)**

Find in a url parameter element in a list.

**Parameters:**

- url\_params* The list of url parameter element to work on.
- name* The name of the parameter element to find.
- dest* A pointer on the element found.

**4.11.3.14 int url\_param\_init (url\_param\_t \*\* *url\_param*)**

Allocate a url parameter element.

**Parameters:**

- url\_param* The element to work on.

**4.11.3.15 int url\_param\_set (url\_param\_t \* *url\_param*, char \* *name*, char \* *value*)**

Set values of a url parameter element.

**Parameters:**

- url\_param* The element to work on.
- name* The token name.
- value* The token value.

**4.11.3.16 int url\_parse (url\_t \* *url*, char \* *buf*)**

Parse a url.

**Parameters:**

- url* The element to work on.
- buf* The buffer to parse.

**4.11.3.17 void url\_sethost (url\_t \* *url*, char \* *value*)**

Set the host of a url element.

**Parameters:**

*url* The element to work on.

*value* The token value.

**4.11.3.18 void url\_setpassword (url\_t \* *url*, char \* *value*)**

Set the password of a url element.

**Parameters:**

*url* The element to work on.

*value* The token value.

**4.11.3.19 void url\_setport (url\_t \* *url*, char \* *value*)**

Set the port of a url element.

**Parameters:**

*url* The element to work on.

*value* The token value.

**4.11.3.20 void url\_setscheme (url\_t \* *url*, char \* *value*)**

Set the scheme of a url element.

**Parameters:**

*url* The element to work on.

*value* The token value.

**4.11.3.21 void url\_setusername (url\_t \* *url*, char \* *value*)**

Set the username of a url element.

**Parameters:**

*url* The element to work on.

*value* The token value.



---

## Chapter 5

# libosip Class Documentation

### 5.1 dialog\_t Struct Reference

```
#include <dialog.h>
```

#### Public Attributes

- char \* **call\_id**
- char \* **local\_tag**
- char \* **remote\_tag**
- list\_t \* **route\_set**
- int **local\_cseq**
- int **remote\_cseq**
- to\_t \* **remote\_uri**
- from\_t \* **local\_uri**
- contact\_t \* **remote\_contact\_uri**
- int **secure**
- dlg\_type\_t **type**
- state\_t **state**

#### 5.1.1 Detailed Description

Structure for referencing a dialog.

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

- **dialog.h**
-



---

## Chapter 6

# libosip File Documentation

### 6.1 dialog.h File Reference

oSIP dialog Routines.

```
#include <osip/osip.h>
```

```
#include <osip/port.h>
```

#### Compounds

- struct **dialog\_t**

#### Typedefs

- typedef dialog\_t **dialog\_t**

#### Functions

- int **dialog\_init\_as\_uac** (dialog\_t \*\*dialog, sip\_t \*response)
- int **dialog\_init\_as\_uas** (dialog\_t \*\*dialog, sip\_t \*invite, sip\_t \*response)
- void **dialog\_free** (dialog\_t \*dialog)
- void **dialog\_set\_state** (dialog\_t \*dialog, dlg\_type\_t type)
- int **dialog\_update\_route\_set\_as\_uas** (dialog\_t \*dialog, sip\_t \*invite)
- int **dialog\_update\_cseq\_as\_uas** (dialog\_t \*dialog, sip\_t \*request)
- int **dialog\_match\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_update\_tag\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_update\_route\_set\_as\_uac** (dialog\_t \*dialog, sip\_t \*response)
- int **dialog\_match\_as\_uas** (dialog\_t \*dialog, sip\_t \*request)

#### 6.1.1 Detailed Description

oSIP dialog Routines.

Dialog management is a powerful facility given by oSIP. This feature is needed by SIP end point who has the capability to answer calls. (i.e. answering 200 OK to an INVITE).

---

A Dialog is a context for a call establishment in oSIP. It's not useless to say that ONE invite request can lead to several call establishment. This can happen if your call has been forked by a proxy and several user agent was contacted and replied at the same time. It is true that this case won't probably happen several times a month...

There is two ways of creating a dialog. In one case, you are the CALLER and in the other case, you will be the CALLEE.

- Creating a dialog as a CALLER

In this case, you have to create a dialog each time you receive an answer with a code between 101 and 299. The best place in oSIP to actually create a dialog is of course in the callback that announce such SIP messages. Of course, each time you receive a response, you have to check for an existing dialog associated to this INVITE that can have been created by earlier SIP answer coming from the same User Agent. The code in the callback will look like the following:

```
void cb_rcv1xx(transaction_t *tr,sip_t *sip)
{
    dialog_t (p.??) *dialog;
    if (MSG_IS_RESPONSEFOR(sip, "INVITE")&&!MSG_TEST_CODE(sip, 100) (p.101))
    {
        dialog = my_application_search_existing_dialog(sip);
        if (dialog==NULL) //NO EXISTING DIALOG
        {
            i = dialog_init_as_uac(&dialog, sip);
            my_application_add_existing_dialog(dialog);
        }
        else {
            // no dialog establishment for other REQUEST
        }
    }
}
```

- Creating a dialog as a CALLEE

In this case, you will have to create a dialog upon receiving the first transmission of the INVITE request. The correct place to do that is inside the callback previously registered to announce new INVITE. First, you will build a SIP answer like 180 or 200 and you'll be able to create a dialog by calling the following code:

```
dialog_t (p.??) *dialog;
dialog_init_as_uas(&dialog, original_invite, response_that_you_build);
```

To make things working, you MUST create a VALID response: do not forget to create a new tag and put it in the 'To' header. The dialog management heavily depends on this tag.

The dialog management is compliant with the latest SIP draft (rfc2543bis-09). It should handle successfully most cases where a remote UA is not compliant (no tag in the To of a final response!) But for example, if you receive 2 answers from 2 uncompliant UA, they will be detected as being related to the same dialog... Do not change any code in oSIP or in your application... instead, you should boycott such implementation. :-



## 6.2 fifo.h File Reference

oSIP fifo Routines.

```
#include <osip/sema.h>
```

```
#include <osip/list.h>
```

### Compounds

- struct **fifo\_t**

### Typedefs

- typedef fifo\_t **fifo\_t**

### Functions

- void **fifo\_init** (fifo\_t \*ff)
- void **fifo\_free** (fifo\_t \*ff)
- int **fifo\_add** (fifo\_t \*ff, void \*element)
- void \* **fifo\_get** (fifo\_t \*ff)
- void \* **fifo\_tryget** (fifo\_t \*ff)

#### 6.2.1 Detailed Description

oSIP fifo Routines.

This is a very simple implementation of a fifo.

There is not much to say about it...

## 6.3 list.h File Reference

oSIP list Routines.

### Compounds

- struct **list\_t**

### Typedefs

- typedef list\_t **list\_t**

### Functions

- int **list\_init** (list\_t \*li)
- void **list\_special\_free** (list\_t \*li, void \*(\*free\_func)(void \*))
- void **listofchar\_free** (list\_t \*li)
- int **list\_size** (list\_t \*li)
- int **list\_eol** (list\_t \*li, int pos)
- int **list\_add** (list\_t \*li, void \*element, int pos)
- void \* **list\_get** (list\_t \*li, int pos)
- int **list\_remove** (list\_t \*li, int pos)

#### 6.3.1 Detailed Description

oSIP list Routines.

This is a very simple implementation of a linked list.

There is not much to say about it... Except that it could be a lot improved. Sadly, it would be difficult to improve it without breaking the compatibility with older version!

## 6.4 osip.h File Reference

oSIP fsm Routines.

```
#include <time.h>
#include <osip/const.h>
#include <osip/msg.h>
#include <osip/fifo.h>
```

### Compounds

- struct **ict\_t**
- struct **ist\_t**
- struct **nict\_t**
- struct **nist\_t**
- struct **osip\_t**
- struct **sipevent\_t**
- struct **transaction\_t**

### Defines

- #define **SIP\_MESSAGE\_MAX\_LENGTH** 4000
- #define **DEFAULT\_T1** 500
- #define **DEFAULT\_T2** 4000
- #define **DEFAULT\_T4** 5000
- #define **EVT\_IS\_RCV\_INVITE**(event) (event → type==RCV\_REQINVITE)
- #define **EVT\_IS\_RCV\_ACK**(event) (event → type==RCV\_REQACK)
- #define **EVT\_IS\_RCV\_REQUEST**(event) (event → type==RCV\_REQUEST)
- #define **EVT\_IS\_RCV\_STATUS\_1XX**(event) (event → type==RCV\_STATUS\_1XX)
- #define **EVT\_IS\_RCV\_STATUS\_2XX**(event) (event → type==RCV\_STATUS\_2XX)
- #define **EVT\_IS\_RCV\_STATUS\_3456XX**(event) (event → type==RCV\_STATUS\_-3456XX)
- #define **EVT\_IS\_SND\_INVITE**(event) (event → type==SND\_REQINVITE)
- #define **EVT\_IS\_SND\_ACK**(event) (event → type==SND\_REQACK)
- #define **EVT\_IS\_SND\_REQUEST**(event) (event → type==SND\_REQUEST)
- #define **EVT\_IS\_SND\_STATUS\_1XX**(event) (event → type==SND\_STATUS\_1XX)
- #define **EVT\_IS\_SND\_STATUS\_2XX**(event) (event → type==SND\_STATUS\_2XX)
- #define **EVT\_IS\_SND\_STATUS\_3456XX**(event) (event → type==SND\_STATUS\_-3456XX)
- #define **EVT\_IS\_INCOMINGMSG**(event)
- #define **EVT\_IS\_INCOMINGREQ**(event)
- #define **EVT\_IS\_INCOMINGRESP**(event)
- #define **EVT\_IS\_OUTGOINGMSG**(event)
- #define **EVT\_IS\_OUTGOINGREQ**(event)
- #define **EVT\_IS\_OUTGOINGRESP**(event)
- #define **EVT\_IS\_MSG**(event)
- #define **EVT\_IS\_KILL\_TRANSACTION**(event) (event → type==KILL\_TRANSACTION)

## Typedefs

- typedef enum **\_state\_t** **state\_t**
- typedef enum type\_t **type\_t**
- typedef enum context\_type\_t **context\_type\_t**
- typedef ict\_t **ict\_t**
- typedef nict\_t **nict\_t**
- typedef ist\_t **ist\_t**
- typedef nist\_t **nist\_t**
- typedef transaction\_t **transaction\_t**
- typedef osip\_t **osip\_t**
- typedef sipevent\_t **sipevent\_t**

## Enumerations

- enum **\_state\_t** { **ICT\_PRE\_CALLING**, **ICT\_CALLING**, **ICT\_PROCEEDING**, **ICT\_COMPLETED**, **ICT\_TERMINATED**, **IST\_PRE\_PROCEEDING**, **IST\_PROCEEDING**, **IST\_COMPLETED**, **IST\_CONFIRMED**, **IST\_TERMINATED**, **NICT\_PRE\_TRYING**, **NICT\_TRYING**, **NICT\_PROCEEDING**, **NICT\_COMPLETED**, **NICT\_TERMINATED**, **NIST\_PRE\_TRYING**, **NIST\_TRYING**, **NIST\_PROCEEDING**, **NIST\_COMPLETED**, **NIST\_TERMINATED** }
- enum **type\_t** { **TIMEOUT\_A**, **TIMEOUT\_B**, **TIMEOUT\_D**, **TIMEOUT\_E**, **TIMEOUT\_F**, **TIMEOUT\_K**, **TIMEOUT\_G**, **TIMEOUT\_H**, **TIMEOUT\_I**, **TIMEOUT\_J**, **RCV\_REQINVITE**, **RCV\_REQACK**, **RCV\_REQUEST**, **RCV\_STATUS\_1XX**, **RCV\_STATUS\_2XX**, **RCV\_STATUS\_3456XX**, **SEND\_REQINVITE**, **SEND\_REQACK**, **SEND\_REQUEST**, **SEND\_STATUS\_1XX**, **SEND\_STATUS\_2XX**, **SEND\_STATUS\_3456XX**, **KILL\_TRANSACTION**, **UNKNOWN\_EVT** }
- enum **context\_type\_t** { **ICT**, **IST**, **NICT**, **NIST** }

## Functions

- int **ict\_set\_destination** (**ict\_t** \*ict, char \*destination, int port)
- int **nict\_set\_destination** (**nict\_t** \*nict, char \*destination, int port)
- **sipevent\_t** \* **nist\_need\_timer\_j\_event** (**nist\_t** \*nist, **state\_t** state, int transactionid)
- int **transaction\_init** (**transaction\_t** \*\*transaction, **context\_type\_t** ctx\_type, **osip\_t** \*osip, **sip\_t** \*request)
- int **transaction\_free** (**transaction\_t** \*transaction)
- int **transaction\_add\_event** (**transaction\_t** \*transaction, **sipevent\_t** \*evt)
- int **transaction\_execute** (**transaction\_t** \*transaction, **sipevent\_t** \*evt)
- int **transaction\_set\_your\_instance** (**transaction\_t** \*transaction, void \*instance)
- void \* **transaction\_get\_your\_instance** (**transaction\_t** \*transaction)
- int **osip\_global\_init** ()
- void **osip\_global\_free** ()
- int **osip\_init** (**osip\_t** \*\*osip)
- void **osip\_free** (**osip\_t** \*osip)
- int **osip\_ict\_execute** (**osip\_t** \*osip)
- int **osip\_ist\_execute** (**osip\_t** \*osip)
- int **osip\_nict\_execute** (**osip\_t** \*osip)
- int **osip\_nist\_execute** (**osip\_t** \*osip)
- void **osip\_timers\_ict\_execute** (**osip\_t** \*osip)

- void `osip_timers_ist_execute` (`osip_t *osip`)
- void `osip_timers_nict_execute` (`osip_t *osip`)
- void `osip_timers_nist_execute` (`osip_t *osip`)
- `transaction_t *` `osip_transaction_find` (`list_t *transactions`, `sipevent_t *evt`)
- `transaction_t *` `osip_find_transaction` (`osip_t *osip`, `sipevent_t *evt`)
- `transaction_t *` `osip_create_transaction` (`osip_t *osip`, `sipevent_t *evt`)
- `sipevent_t *` `osip_parse` (`char *buf`)
- `sipevent_t *` `osip_new_outgoing_sipmessage` (`sip_t *sip`)
- void `osip_setcb_send_message` (`osip_t *cf`, `int(*cb)(transaction_t *, sip_t *, char *, int, int)`)
- void `osip_setcb_ict_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_ict_invite_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_invite_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_ack_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_ack_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_1xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_2xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_2xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_3xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_4xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_5xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_6xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_3456xx_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ict_transport_error` (`osip_t *cf`, `void(*cb)(transaction_t *, int error)`)
- void `osip_setcb_ist_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_ist_invite_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_invite_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_ack_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_ack_received2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_1xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_1xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_2xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_2xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_3xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_4xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_5xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_6xx_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_3456xx_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_ist_transport_error` (`osip_t *cf`, `void(*cb)(transaction_t *, int error)`)
- void `osip_setcb_nict_kill_transaction` (`osip_t *cf`, `void(*cb)(transaction_t *)`)
- void `osip_setcb_nict_register_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_bye_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_options_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_info_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_cancel_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_notify_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_subscribe_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_unknown_sent` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_request_sent2` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)
- void `osip_setcb_nict_1xx_received` (`osip_t *cf`, `void(*cb)(transaction_t *, sip_t *)`)

- void `osip_setcb_nict_2xx_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_2xx_received2` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_3xx_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_4xx_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_5xx_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_6xx_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_3456xx_received2` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nict_transport_error` (`osip_t *cf`, void(\*cb)(`transaction_t *`, int error))
- void `osip_setcb_nist_kill_transaction` (`osip_t *cf`, void(\*cb)(`transaction_t *`))
- void `osip_setcb_nist_register_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_bye_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_options_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_info_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_cancel_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_notify_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_subscribe_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_unknown_received` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_request_received2` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_1xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_2xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_2xx_sent2` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_3xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_4xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_5xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_6xx_sent` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_3456xx_sent2` (`osip_t *cf`, void(\*cb)(`transaction_t *`, `sip_t *`))
- void `osip_setcb_nist_transport_error` (`osip_t *cf`, void(\*cb)(`transaction_t *`, int error))

### 6.4.1 Detailed Description

oSIP fsm Routines.

#### Introduction.

fsm stands for 'finite state machine'. The possible STATE of the state machines are defined in the enum state. In oSIP, you can actually find 4 different state machines. Those state machines definitions are directly related to the definitions of transactions from the SIP specifications. (See section: 17.1.1, 17.1.2, 17.2.1, 17.2.2). In the 4 drawings shown in those sections, you'll find the possible STATES and the possible EVENTS (`sipevent_t`) that can occur. EVENTS can be either TIMEOUT events and SIP message (incoming and outgoing) events.

#### Why 4 finite state machines.

SIP has two different kind of transaction: INVITE and NON-INVITE ones. Also, a SIP User Agent can act as a server and as a client. This simply leads to 4 transactions state machines.

## Step 1: oSIP initialisation

To use oSIP, a program **MUST** first initialise internal elements in the stack. The initialisation is shown below:

```

osip_t *osip;
// initialise internal element first
if (0!= osip`global`init() (p.22))
    return -1;
// allocate a global osip element.
if (0!=osip_init(&osip))
    return -1;

// the next step is the initialisation of the callbacks used by the
// oSIP stack to announce events (when a transition occurs in the fsm)

// This callback is somewhat special and is used by oSIP to inform
// the application that a message has to be sent. The message is
// sent by your application! oSIP has no ways to send it alone.
// Also, the method you supply will be called with default values where
// you should send the SIP message. You are not mandated to send the
// SIP message by using those default values.
// the callback MUST return 0 on success, 1 on ECONNREFUSED, -1 on error.
osip_setcb_send_message(osip, &application_cb_snd_message);

// here is the long list of callback that you can register. Some
// of this callbacks are very useless (announcing a retransmission,
// or announcing that you have sent a SIP message which you may already
// know...).

// those callbacks are mandatory. They are called when oSIP has decided
// that this transaction MUST no longer be handled by oSIP. (This is
// called in both successful or error cases scenario)
osip_setcb_ict_kill_transaction(osip,&application_cb_ict_kill_transaction);
osip_setcb_ist_kill_transaction(osip,&application_cb_ist_kill_transaction);
osip_setcb_nict_kill_transaction(osip,&application_cb_nict_kill_transaction);
osip_setcb_nist_kill_transaction(osip,&application_cb_nist_kill_transaction);

// those callbacks are optional. The purpose is to announce retransmissions
// of SIP message decided by the oSIP stack. (They can be used for statistics?)
osip_setcb_ict_2xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_ict_3456xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_ict_invite_sent2(osip,&application_cb_sndreq_retransmission);
osip_setcb_ist_2xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_ist_3456xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_ist_invite_received2(osip,&application_cb_rcvreq_retransmission);
osip_setcb_nict_2xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_nict_3456xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_nict_request_sent2(osip,&application_cb_sndreq_retransmission);
osip_setcb_nist_2xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_nist_3456xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_nist_request_received2(osip,&application_cb_rcvreq_retransmission);

// those callbacks are mandatory. They are used to announce network related
// errors (the return code of the network callback if it was not 0)
osip_setcb_ict_transport_error(osip,&application_cb_transport_error);

```

```
osip_setcb_ist_transport_error(osip,&application_cb_transport_error);
osip_setcb_nict_transport_error(osip,&application_cb_transport_error);
osip_setcb_nist_transport_error(osip,&application_cb_transport_error);

// those callbacks are optional. They are used to announce the initial
// request sent for a newly created transaction.
osip_setcb_ict_invite_sent (osip,&application_cb_sndinvite);
osip_setcb_ict_ack_sent    (osip,&application_cb_sndack);
osip_setcb_nict_register_sent(osip,&application_cb_sndregister);
osip_setcb_nict_bye_sent    (osip,&application_cb_sndbye);
osip_setcb_nict_cancel_sent (osip,&application_cb_sndcancel);
osip_setcb_nict_info_sent   (osip,&application_cb_sndinfo);
osip_setcb_nict_options_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_subscribe_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_notify_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_unknown_sent(osip,&application_cb_sndunkrequest);

// those callbacks are mandatory. They are used to announce the initial
// response received for a transaction. (for SIP response between 100 and 199,
// all responses are announced because this is not a retransmission case)
osip_setcb_ict_1xx_received(osip,&application_cb_rcv1xx);
osip_setcb_ict_2xx_received(osip,&application_cb_rcv2xx);
osip_setcb_ict_3xx_received(osip,&application_cb_rcv3xx);
osip_setcb_ict_4xx_received(osip,&application_cb_rcv4xx);
osip_setcb_ict_5xx_received(osip,&application_cb_rcv5xx);
osip_setcb_ict_6xx_received(osip,&application_cb_rcv6xx);

// those callbacks are optional. They are used to announce the initial
// response sent for a transaction. (for SIP response between 100 and 199,
// all responses are announced because this is not a retransmission case)
osip_setcb_ist_1xx_sent(osip,&application_cb_snd1xx);
osip_setcb_ist_2xx_sent(osip,&application_cb_snd2xx);
osip_setcb_ist_3xx_sent(osip,&application_cb_snd3xx);
osip_setcb_ist_4xx_sent(osip,&application_cb_snd4xx);
osip_setcb_ist_5xx_sent(osip,&application_cb_snd5xx);
osip_setcb_ist_6xx_sent(osip,&application_cb_snd6xx);

// those callbacks are mandatory. They are used to announce the initial
// response received for a transaction. (for SIP response between 100 and 199,
// all responses are announced because this is not a retransmission case)
osip_setcb_nict_1xx_received(osip,&application_cb_rcv1xx);
osip_setcb_nict_2xx_received(osip,&application_cb_rcv2xx);
osip_setcb_nict_3xx_received(osip,&application_cb_rcv3xx);
osip_setcb_nict_4xx_received(osip,&application_cb_rcv4xx);
osip_setcb_nict_5xx_received(osip,&application_cb_rcv5xx);
osip_setcb_nict_6xx_received(osip,&application_cb_rcv6xx);

// those callbacks are optional. They are used to announce the initial
// response sent for a transaction. (for SIP response between 100 and 199,
// all responses are announced because this is not a retransmission case)
osip_setcb_nist_1xx_sent(osip,&application_cb_snd1xx);
osip_setcb_nist_2xx_sent(osip,&application_cb_snd2xx);
osip_setcb_nist_3xx_sent(osip,&application_cb_snd3xx);
osip_setcb_nist_4xx_sent(osip,&application_cb_snd4xx);
osip_setcb_nist_5xx_sent(osip,&application_cb_snd5xx);
osip_setcb_nist_6xx_sent(osip,&application_cb_snd6xx);
```



```

// those callbacks are mandatory. They are used to announce the initial
// request received for a transaction. It is not useless to notice that
// a special behaviour exist for the 200 OK and the ACK in the case of
// a successful INVITE transaction. This will be discussed later.
osip_setcb_ist_invite_received (osip,&application_cb_rcvinvoke);
osip_setcb_ist_ack_received (osip,&application_cb_rcvack);
// this callback is optional
osip_setcb_ist_ack_received2 (osip,&application_cb_rcvack2);
osip_setcb_nist_register_received(osip,&application_cb_rcvregister);
osip_setcb_nist_bye_received (osip,&application_cb_rcvbye);
osip_setcb_nist_cancel_received (osip,&application_cb_rcvcancel);
osip_setcb_nist_info_received (osip,&application_cb_rcvinfo);
osip_setcb_nist_options_received (osip,&application_cb_rcvoptions);
osip_setcb_nist_subscribe_received(osip,&application_cb_rcvoptions);
osip_setcb_nist_notify_received (osip,&application_cb_rcvoptions);
osip_setcb_nist_unknown_received (osip,&application_cb_rcvunkrequest);

```

## Step 2: Initialising a new transaction.

Let's assume you want to implement a User Agent and you want to start a REGISTER transaction. Using the parser library, you will first have to build a SIP compliant message. (oSIP, as a low layer library provides an interface to build SIP messages, but it's up to you to correctly fill all the required fields.) As soon as you have build the SIP message, you are ready to start a new transaction. Here is the code:

```

osip_t *osip = your_global_osip_context;
transaction_t *transaction;
sip_t *sip_register_message;
sipevent_t *sipevent;

application_build_register(&sip_register_message);
transaction_init(&transaction,
    NICT, //a REGISTER is a Non-Invite-Client-Transaction
    osip,
    sip_register_message);

// If you have a special context that you want to associate to that
// transaction, you can use a special method that associate your context
// to the transaction context.

transaction_set_your_instance(transaction, my_context);

// at this point, the transaction context exists in oSIP but you still have
// to give the SIP message to the finite state machine.
sipevent = osip_new_outgoing_sipmessage(msg);
sipevent->transactionid = transaction->transactionid;
transaction_add_event(transaction, sipevent);
// at this point, the event will be handled by oSIP. (The memory resource will
// also be handled by oSIP). Note that no action is taken there.

```

Adding new events in the fsm is made with similar code.

### Step 3: Consuming events.

The previous step show how to create a transaction and one possible way to add a new event. (Note, that some events -the TIMEOUT\_\* ones- will be added by oSIP not by the application). In this step, we describe how the oSIP stack will consume events. In fact, this is very simple, but you should be aware that it's not always allowed to consume an event at any time! The fsm **MUST** consume events sequentially within a transaction. This means that when your are calling **transaction\_execute()** (p.37), it is forbidden to call this method again with the same transaction context until the first call has returned. In a multi threaded application, if one thread handles one transaction, the code will be the following:

```
while (1)
{
    se = (sipevent_t *)fifo_get(transaction->transactionff);
    if (se==NULL)
        pthread_exit() (p.??);
    if (transaction_execute(transaction,se)<1) // deletion asked
        pthread_exit() (p.??);
}
```

### Step 4: How the stack will announce the events

Looking at the case of a usual outgoing REGISTER transaction, this behaviour is expected.

When an event is seen as useful for the fsm, it means that a transition from one state to another has to be done on the transaction context. If the event is SND\_REQUEST (this is the case for an outgoing REGISTER), the callback previously registered to announce this action will be called. This callback is useless for the application as no action has to be taken at this step. A more interesting announcement will be made when consuming the first final response received. If the callbacks associated to 2xx message is called, then the transaction has succeeded. Inside this callback, you will probably inform the user of the success of the registration if you want to do so... If the final response is not a 2xx, or the network callback is called, you'll probably want to take some actions. For example, if you receive a 302, you'll probably want to retry a registration at the new location. All that decision is up to you.

## 6.5 sdp.h File Reference

oSIP SDP parser Routines.

```
#include <osip/list.h>
```

### Compounds

- struct **sdp\_attribute\_t**
- struct **sdp\_bandwidth\_t**
- struct **sdp\_connection\_t**
- struct **sdp\_key\_t**
- struct **sdp\_media\_t**
- struct **sdp\_t**
- struct **sdp\_time\_descr\_t**

### Typedefs

- typedef sdp\_bandwidth\_t **sdp\_bandwidth\_t**
- typedef sdp\_time\_descr\_t **sdp\_time\_descr\_t**
- typedef sdp\_key\_t **sdp\_key\_t**
- typedef sdp\_attribute\_t **sdp\_attribute\_t**
- typedef sdp\_connection\_t **sdp\_connection\_t**
- typedef sdp\_media\_t **sdp\_media\_t**
- typedef sdp\_t **sdp\_t**

### Functions

- int **sdp\_bandwidth\_init** (sdp\_bandwidth\_t \*\*elem)
- void **sdp\_bandwidth\_free** (sdp\_bandwidth\_t \*elem)
- int **sdp\_time\_descr\_init** (sdp\_time\_descr\_t \*\*elem)
- void **sdp\_time\_descr\_free** (sdp\_time\_descr\_t \*elem)
- int **sdp\_key\_init** (sdp\_key\_t \*\*elem)
- void **sdp\_key\_free** (sdp\_key\_t \*elem)
- int **sdp\_attribute\_init** (sdp\_attribute\_t \*\*elem)
- void **sdp\_attribute\_free** (sdp\_attribute\_t \*elem)
- int **sdp\_connection\_init** (sdp\_connection\_t \*\*elem)
- void **sdp\_connection\_free** (sdp\_connection\_t \*elem)
- int **sdp\_media\_init** (sdp\_media\_t \*\*elem)
- void **sdp\_media\_free** (sdp\_media\_t \*elem)
- int **sdp\_init** (sdp\_t \*\*sdp)
- int **sdp\_parse** (sdp\_t \*sdp, const char \*buf)
- int **sdp\_2char** (sdp\_t \*sdp, char \*\*dest)
- void **sdp\_free** (sdp\_t \*sdp)
- int **sdp\_v\_version\_set** (sdp\_t \*sdp, char \*value)
- char \* **sdp\_v\_version\_get** (sdp\_t \*sdp)
- int **sdp\_o\_origin\_set** (sdp\_t \*sdp, char \*username, char \*sess\_id, char \*sess\_version, char \*nettype, char \*addrtype, char \*addr)
- char \* **sdp\_o\_username\_get** (sdp\_t \*sdp)
- char \* **sdp\_o\_sess\_id\_get** (sdp\_t \*sdp)

- `char * sdp_o_sess_version_get (sdp_t *sdp)`
- `char * sdp_o_nettype_get (sdp_t *sdp)`
- `char * sdp_o_addrtype_get (sdp_t *sdp)`
- `char * sdp_o_addr_get (sdp_t *sdp)`
- `int sdp_s_name_set (sdp_t *sdp, char *value)`
- `char * sdp_s_name_get (sdp_t *sdp)`
- `int sdp_i_info_set (sdp_t *sdp, int pos_media, char *value)`
- `char * sdp_i_info_get (sdp_t *sdp, int pos_media)`
- `int sdp_u_uri_set (sdp_t *sdp, char *value)`
- `char * sdp_u_uri_get (sdp_t *sdp)`
- `int sdp_e_email_add (sdp_t *sdp, char *value)`
- `char * sdp_e_email_get (sdp_t *sdp, int pos)`
- `int sdp_p_phone_add (sdp_t *sdp, char *value)`
- `char * sdp_p_phone_get (sdp_t *sdp, int pos)`
- `int sdp_c_connection_add (sdp_t *sdp, int pos_media, char *nettype, char *addrtype, char *addr, char *addr_multicast_ttl, char *addr_multicast_int)`
- `char * sdp_c_nettype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addrtype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_multicast_ttl_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_c_addr_multicast_int_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_b_bandwidth_add (sdp_t *sdp, int pos_media, char *bwtype, char *bandwidth)`
- `sdp_bandwidth_t * sdp_bandwidth_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_b_bwtype_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_b_bandwidth_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_t_time_descr_add (sdp_t *sdp, char *start, char *stop)`
- `char * sdp_t_start_time_get (sdp_t *sdp, int pos_td)`
- `char * sdp_t_stop_time_get (sdp_t *sdp, int pos_td)`
- `int sdp_r_repeat_add (sdp_t *sdp, int pos_time_descr, char *value)`
- `char * sdp_r_repeat_get (sdp_t *sdp, int pos_time_descr, int pos_repeat)`
- `int sdp_z_adjustments_set (sdp_t *sdp, char *value)`
- `char * sdp_z_adjustments_get (sdp_t *sdp)`
- `int sdp_k_key_set (sdp_t *sdp, int pos_media, char *keytype, char *keydata)`
- `char * sdp_k_keytype_get (sdp_t *sdp, int pos_media)`
- `char * sdp_k_keydata_get (sdp_t *sdp, int pos_media)`
- `int sdp_a_attribute_add (sdp_t *sdp, int pos_media, char *att_field, char *att_value)`
- `sdp_attribute_t * sdp_attribute_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_a_att_field_get (sdp_t *sdp, int pos_media, int pos)`
- `char * sdp_a_att_value_get (sdp_t *sdp, int pos_media, int pos)`
- `int sdp_endof_media (sdp_t *sdp, int pos)`
- `int sdp_m_media_add (sdp_t *sdp, char *media, char *port, char *number_of_port, char *proto)`
- `char * sdp_m_media_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_port_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_number_of_port_get (sdp_t *sdp, int pos_media)`
- `char * sdp_m_proto_get (sdp_t *sdp, int pos_media)`
- `int sdp_m_payload_add (sdp_t *sdp, int pos_media, char *payload)`
- `char * sdp_m_payload_get (sdp_t *sdp, int pos_media, int pos)`

### 6.5.1 Detailed Description

oSIP SDP parser Routines.

This is the SDP accessor and parser related API.

## 6.6 sdp\_negoc.h File Reference

oSIP and SDP offer/answer model Routines.

```
#include <osip/sdp.h>
```

### Compounds

- struct **payload\_t**
- struct **sdp\_config\_t**
- struct **sdp\_context\_t**

### Typedefs

- typedef sdp\_context\_t **sdp\_context\_t**
- typedef payload\_t **payload\_t**
- typedef sdp\_config\_t **sdp\_config\_t**

### Functions

- int **sdp\_context\_init** (sdp\_context\_t \*\*ctx)
- void **sdp\_context\_free** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_mycontext** (sdp\_context\_t \*ctx, void \*value)
- void \* **sdp\_context\_get\_mycontext** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_local\_sdp** (sdp\_context\_t \*ctx, sdp\_t \*sdp)
- sdp\_t \* **sdp\_context\_get\_local\_sdp** (sdp\_context\_t \*ctx)
- int **sdp\_context\_set\_remote\_sdp** (sdp\_context\_t \*ctx, sdp\_t \*sdp)
- sdp\_t \* **sdp\_context\_get\_remote\_sdp** (sdp\_context\_t \*ctx)
- int **payload\_init** (payload\_t \*\*payload)
- void **payload\_free** (payload\_t \*payload)
- int **sdp\_config\_init** ()
- void **sdp\_config\_free** ()
- int **sdp\_config\_set\_o\_username** (char \*tmp)
- int **sdp\_config\_set\_o\_session\_id** (char \*tmp)
- int **sdp\_config\_set\_o\_session\_version** (char \*tmp)
- int **sdp\_config\_set\_o\_nettype** (char \*tmp)
- int **sdp\_config\_set\_o\_addrtype** (char \*tmp)
- int **sdp\_config\_set\_o\_addr** (char \*tmp)
- int **sdp\_config\_set\_c\_nettype** (char \*tmp)
- int **sdp\_config\_set\_c\_addrtype** (char \*tmp)
- int **sdp\_config\_set\_c\_addr** (char \*tmp)
- int **sdp\_config\_set\_c\_addr\_multicast\_ttl** (char \*tmp)
- int **sdp\_config\_set\_c\_addr\_multicast\_int** (char \*tmp)
- int **sdp\_config\_add\_support\_for\_audio\_codec** (char \*payload, char \*number\_of\_port, char \*proto, char \*c\_nettype, char \*c\_addrtype, char \*c\_addr, char \*c\_addr\_multicast\_ttl, char \*c\_addr\_multicast\_int, char \*a\_rtpmap)
- int **sdp\_config\_add\_support\_for\_video\_codec** (char \*payload, char \*number\_of\_port, char \*proto, char \*c\_nettype, char \*c\_addrtype, char \*c\_addr, char \*c\_addr\_multicast\_ttl, char \*c\_addr\_multicast\_int, char \*a\_rtpmap)

- `int sdp_config_add_support_for_other_codec` (`char *payload`, `char *number_of_port`, `char *proto`, `char *c_nettype`, `char *c_addrtype`, `char *c_addr`, `char *c_addr_multicast_ttl`, `char *c_addr_multicast_int`, `char *a_rtpmap`)
- `int sdp_config_set_fcn_set_info` (`int(*fcn)(sdp_context_t *, sdp_t *)`)
- `int sdp_config_set_fcn_set_uri` (`int(*fcn)(sdp_context_t *, sdp_t *)`)
- `int sdp_config_set_fcn_set_emails` (`int(*fcn)(sdp_context_t *, sdp_t *)`)
- `int sdp_config_set_fcn_set_phones` (`int(*fcn)(sdp_context_t *, sdp_t *)`)
- `int sdp_config_set_fcn_set_attributes` (`int(*fcn)(sdp_context_t *, sdp_t *, int)`)
- `int sdp_config_set_fcn_accept_audio_codec` (`int(*fcn)(sdp_context_t *, char *, char *, int, char *)`)
- `int sdp_config_set_fcn_accept_video_codec` (`int(*fcn)(sdp_context_t *, char *, char *, int, char *)`)
- `int sdp_config_set_fcn_accept_other_codec` (`int(*fcn)(sdp_context_t *, char *, char *, char *, char *)`)
- `int sdp_config_set_fcn_get_audio_port` (`char *(*fcn)(sdp_context_t *, int)`)
- `int sdp_config_set_fcn_get_video_port` (`char *(*fcn)(sdp_context_t *, int)`)
- `int sdp_config_set_fcn_get_other_port` (`char *(*fcn)(sdp_context_t *, int)`)
- `int sdp_context_execute_negotiation` (`sdp_context_t *ctx`)

### 6.6.1 Detailed Description

oSIP and SDP offer/answer model Routines.

The SDP offer/answer model is where most SIP interoperability issue comes from. The SDP specification (rfc2327.txt) is often not fully respected. As an example, most SIP applications forget to add the mandatory 's' field in the SDP packet. Another mistake is to assume that an SDP packet don't need a 'p' and a 'e' field. Even if they are both optional, at least of those is mandatory! I have never seen ONE implementation that send at least one 'p' or 'e' field!!

For all the reasons, that make negotiation a hard task, I have decided to provide a helpful facility to build SDP answer from an SDP offer. (This facility does not help to build the compliant offer) Of course, after the SDP negotiator has been executed and produced a valid response, you can still modify your SDP answer to add attributes or modify anything. You always keep the entire control over it.

### Do you need the negotiator

If you are planning a simple application, I advise you to use it. Advanced applications may find it inappropriate, but as you can modify the SDP answer after running the negotiation, I see no reason why you should not use it. The only goal of the SDP negotiator is to make sure only one line of audio codec is accepted (the first one) and only one line of video codec is accepted (the first one). It also remove from the media lines, the codec that you don't support without asking you. (Also, you can still refuse the codec you support.)

Using the negotiator, your only task is to check/add/remove the media attributes.

### How-To

Using the SDP negotiator is simple. An example is provided in the test directory as 'torture\_sdp.c'. It parses a SDP packet from a file (a sample is available in conf/) and produce the answer that would be made with a basic configuration where 4 audio codecs are supported.

When starting your application, you simply configure the global `sdp_config_t` element: you'll set you username, ip address and some general informations about you that every SDP packet must contain. As a second action, you will register all the codec you support. (audio, video and 'other' codecs).

After that, you will also register a set of method used to accept the codec. The return code of those method will accept or refused the supported codec for this specific session.

```
sdp_config_set_fcn_accept_audio_codec(&application_accept_audio_codec);
sdp_config_set_fcn_accept_video_codec(&application_accept_video_codec);
sdp_config_set_fcn_accept_other_codec(&application_accept_other_codec);
sdp_config_set_fcn_get_audio_port(&application_get_audio_port);
```

When you need to create an answer, the following code will create the SDP packet:

```
sdp_context_t *context;

sdp_t *dest;
i = sdp_context_init(&context);
i = sdp_context_set_mycontext(context, (void *)ua_context);
i = sdp_context_set_remote_sdp(context, sdp);
if (i!=0) {
    fprintf(stdout, "Initialisation of context failed. Could not negotiate");
} else {
    fprintf(stdout, "Trying to execute a SIP negociation:");
    i = sdp_context_execute_negociation(context);
    fprintf(stdout, "return code: i",i);
    if (i==200)
    {
        dest = sdp_context_get_local_sdp(context);
        fprintf(stdout, "SDP answer:");
        i = sdp_2char(dest, &result);
        if (i!=0)
            fprintf(stdout, "Error found in SDP answer while printing");
        else
            fprintf(stdout, "s", result);
        sfree(result);
    }
    sdp_context_free(context);
    sfree(context);
    return 0;
}
```

Notice the presence of `sdp_context_set_mycontext()` (p.62) which can add a store the address of your own context (probably related to your call). This is very useful if you need to know inside the callback which call this negotiation belongs to.



## 6.7 sema.h File Reference

oSIP semaphore definitions.

```
#include <osip/port.h>
```

```
#include <errno.h>
```

```
#include <pthread.h>
```

```
#include <semaphore.h>
```

### Typedefs

- typedef pthread\_mutex\_t **smutex\_t**
- typedef sem\_t **ssem\_t**

### Functions

- **smutex\_t** \* **smutex\_init** ()
- void **smutex\_destroy** (**smutex\_t** \*mut)
- int **smutex\_lock** (**smutex\_t** \*mut)
- int **smutex\_unlock** (**smutex\_t** \*mut)
- **ssem\_t** \* **ssem\_init** (unsigned int value)
- int **ssem\_destroy** (**ssem\_t** \*sem)
- int **ssem\_post** (**ssem\_t** \*sem)
- int **ssem\_wait** (**ssem\_t** \*sem)
- int **ssem\_trywait** (**ssem\_t** \*sem)

#### 6.7.1 Detailed Description

oSIP semaphore definitions.

Those methods are only available if the library is compile in multi threaded mode. This is the default for oSIP.

## 6.8 smsg.h File Reference

oSIP parser Routines.

```
#include <osip/const.h>
```

```
#include <osip/smsgtypes.h>
```

### Defines

- `#define MSG_IS_RESPONSE(msg) ((msg) → strtline → statuscode!=NULL)`
- `#define MSG_IS_REQUEST(msg) ((msg) → strtline → statuscode==NULL)`
- `#define MSG_IS_INVITE(msg)`
- `#define MSG_IS_ACK(msg)`
- `#define MSG_IS_REGISTER(msg)`
- `#define MSG_IS_BYE(msg)`
- `#define MSG_IS_OPTIONS(msg)`
- `#define MSG_IS_INFO(msg)`
- `#define MSG_IS_CANCEL(msg)`
- `#define MSG_IS_NOTIFY(msg)`
- `#define MSG_IS_SUBSCRIBE(msg)`
- `#define MSG_IS_PRACK(msg)`
- `#define MSG_IS_STATUS_1XX(msg)`
- `#define MSG_IS_STATUS_2XX(msg)`
- `#define MSG_IS_STATUS_3XX(msg)`
- `#define MSG_IS_STATUS_4XX(msg)`
- `#define MSG_IS_STATUS_5XX(msg)`
- `#define MSG_IS_STATUS_6XX(msg)`
- `#define MSG_TEST_CODE(msg, code)`
- `#define MSG_IS_RESPONSEFOR(msg, requestname)`
- `#define generic_param_init(GP) url_param_init(GP)`
- `#define generic_param_free(GP) url_param_free(GP)`
- `#define generic_param_set(GP, NAME, VALUE) url_param_set(GP, NAME, VALUE)`
- `#define generic_param_clone(GP, DEST) url_param_clone(GP,DEST)`
- `#define generic_param_add(LIST, NAME, VALUE) url_param_-  
add(LIST,NAME,VALUE)`
- `#define generic_param_getbyname(LIST, NAME, DEST) url_param_-  
getbyname(LIST,NAME,DEST)`
- `#define accept_init(header) content_type_init(header)`
- `#define accept_free(header) content_type_free(header)`
- `#define accept_parse(header, hvalue) content_type_parse(header, hvalue)`
- `#define accept_2char(header, dest) content_type_2char(header, dest)`
- `#define accept_clone(header, dest) content_type_clone(header, dest)`
- `#define accept_param_get(header, pos, dest) generic_param_get((header) → gen_params,  
pos, dest)`
- `#define accept_param_add(header, name, value) generic_param_add((header) → gen_-  
params,name,value)`
- `#define accept_param_getbyname(header, name, dest) generic_param_-  
getbyname((header) → gen_params,name,dest)`
- `#define accept_encoding_param_get(header, pos, dest) generic_param_get((header) →  
gen_params, pos, dest)`

- `#define accept_encoding_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define accept_encoding_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define accept_language_init(header) accept_encoding_init(header)`
- `#define accept_language_parse(header, hvalue) accept_encoding_parse(header, hvalue)`
- `#define accept_language_2char(header, dest) accept_encoding_2char(header, dest)`
- `#define accept_language_free(header) accept_encoding_free(header)`
- `#define accept_language_clone(header, dest) accept_encoding_clone(header, dest)`
- `#define accept_language_getelement(header) accept_encoding_getelement(header)`
- `#define accept_language_setelement(header, value) accept_encoding_setelement(header, value)`
- `#define accept_language_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos,dest)`
- `#define accept_language_param_add(header, name, value) generic_param_add((header) → gen_params,name,value)`
- `#define accept_language_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define alert_info_init(header) call_info_init(header)`
- `#define alert_info_free(header) call_info_free(header)`
- `#define alert_info_parse(header, hvalue) call_info_parse(header, hvalue)`
- `#define alert_info_2char(header, dest) call_info_2char(header,dest)`
- `#define alert_info_clone(header, dest) call_info_clone(header, dest)`
- `#define alert_info_geturi(header) call_info_geturi(header)`
- `#define alert_info_seturi(header, uri) call_info_seturi(header, uri)`
- `#define allow_init(header) content_length_init(header)`
- `#define allow_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define allow_2char(header, dest) content_length_2char(header, dest)`
- `#define allow_free(header) content_length_free(header)`
- `#define allow_clone(header, dest) content_length_clone(header, dest)`
- `#define contact_getdisplayname(header) from_getdisplayname((from_t*)header)`
- `#define contact_setdisplayname(header, value) from_setdisplayname((from_t*)header, value)`
- `#define contact_geturl(header) from_geturl((from_t*)header)`
- `#define contact_seturl(header, url) from_seturl((from_t*)header,url)`
- `#define contact_param_get(header, pos, dest) from_param_get((from_t*)header,pos,dest)`
- `#define contact_param_add(header, name, value) generic_param_add((header) → gen_params, name,value)`
- `#define contact_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params,name,dest)`
- `#define content_disposition_init(header) call_info_init(header)`
- `#define content_disposition_free(header) call_info_free(header)`
- `#define content_disposition_2char(header, dest) call_info_2char(header,dest)`
- `#define content_disposition_clone(header, dest) call_info_clone(header, dest)`
- `#define content_disposition_settype(header, value) call_info_seturi(header, value)`
- `#define content_disposition_gettype(header) call_info_geturi(header)`
- `#define content_encoding_init(header) content_length_init(header)`
- `#define content_encoding_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define content_encoding_2char(header, dest) content_length_2char(header, dest)`
- `#define content_encoding_free(header) content_length_free(header)`
- `#define content_encoding_clone(header, dest) content_length_clone(header, dest)`

- `#define content_type_param_get(header, pos, dest) generic_param_get((header) → gen_params, pos, dest)`
- `#define content_type_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define content_type_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`
- `#define error_info_init(header) call_info_init(header)`
- `#define error_info_free(header) call_info_free(header)`
- `#define error_info_parse(header, hvalue) call_info_parse(header, hvalue)`
- `#define error_info_2char(header, dest) call_info_2char(header, dest)`
- `#define error_info_clone(header, dest) call_info_clone(header, dest)`
- `#define error_info_seturi(header, uri) call_info_seturi(header, uri)`
- `#define error_info_geturi(header) call_info_geturi(header)`
- `#define from_param_add(header, name, value) generic_param_add((header) → gen_params, name, value)`
- `#define from_param_getbyname(header, name, dest) generic_param_getbyname((header) → gen_params, name, dest)`
- `#define from_get_tag(header, dest) generic_param_getbyname((header) → gen_params, "tag", dest)`
- `#define from_set_tag(header, value) generic_param_add((header) → gen_params, sget_copy("tag"), value)`
- `#define mime_version_init(header) content_length_init(header)`
- `#define mime_version_parse(header, hvalue) content_length_parse(header, hvalue)`
- `#define mime_version_2char(header, dest) content_length_2char(header, dest)`
- `#define mime_version_free(header) content_length_free(header)`
- `#define mime_version_clone(header, dest) content_length_clone(header, dest)`
- `#define proxy_authenticate_init(header) www_authenticate_init(header)`
- `#define proxy_authenticate_parse(header, hvalue) www_authenticate_parse(header, hvalue)`
- `#define proxy_authenticate_2char(header, dest) www_authenticate_2char(header, dest)`
- `#define proxy_authenticate_free(header) www_authenticate_free(header)`
- `#define proxy_authenticate_clone(header, dest) www_authenticate_clone(header, dest)`
- `#define proxy_authenticate_getauth_type(header) www_authenticate_getauth_type(header)`
- `#define proxy_authenticate_setauth_type(header, value) www_authenticate_setauth_type(header, value)`
- `#define proxy_authenticate_getrealm(header) www_authenticate_getrealm(header)`
- `#define proxy_authenticate_setrealm(header, value) www_authenticate_setrealm(header, value)`
- `#define proxy_authenticate_getdomain(header) www_authenticate_getdomain(header)`
- `#define proxy_authenticate_setdomain(header, value) www_authenticate_setdomain(header, value)`
- `#define proxy_authenticate_getnonce(header) www_authenticate_getnonce(header)`
- `#define proxy_authenticate_setnonce(header, value) www_authenticate_setnonce(header, value)`
- `#define proxy_authenticate_getopaque(header) www_authenticate_getopaque(header)`
- `#define proxy_authenticate_setopaque(header, value) www_authenticate_setopaque(header, value)`
- `#define proxy_authenticate_getstale(header) www_authenticate_getstale(header)`
- `#define proxy_authenticate_setstale(header, value) www_authenticate_setstale(header, value)`

- `#define proxy_authenticate_setstale_true(header) www_authenticate_setstale(header, sgetcopy("true"))`
- `#define proxy_authenticate_setstale_false(header) www_authenticate_setstale(header, sgetcopy("false"))`
- `#define proxy_authenticate_getalgorithm(header) www_authenticate_getalgorithm(header)`
- `#define proxy_authenticate_setalgorithm(header, value) www_authenticate_setalgorithm(header, value)`
- `#define proxy_authenticate_setalgorithm_MD5(header) www_authenticate_setalgorithm(header, sgetcopy("MD5"))`
- `#define proxy_authenticate_getqop_options(header) www_authenticate_getqop_options(header)`
- `#define proxy_authenticate_setqop_options(header, value) www_authenticate_setqop_options(header, value)`
- `#define proxy_authorization_init(header) authorization_init(header)`
- `#define proxy_authorization_parse(header, hvalue) authorization_parse(header, hvalue)`
- `#define proxy_authorization_2char(header, dest) authorization_2char(header, dest)`
- `#define proxy_authorization_free(header) authorization_free(header)`
- `#define proxy_authorization_clone(header, dest) authorization_clone(header, dest)`
- `#define proxy_authorization_getauth_type(header) authorization_getauth_type(header)`
- `#define proxy_authorization_setauth_type(header, value) authorization_setauth_type(header, value)`
- `#define proxy_authorization_getusername(header) authorization_getusername(header)`
- `#define proxy_authorization_setusername(header, value) authorization_setusername(header, value)`
- `#define proxy_authorization_getrealm(header) authorization_getrealm(header)`
- `#define proxy_authorization_setrealm(header, value) authorization_setrealm(header, value)`
- `#define proxy_authorization_getnonce(header) authorization_getnonce(header)`
- `#define proxy_authorization_setnonce(header, value) authorization_setnonce(header, value)`
- `#define proxy_authorization_geturi(header) authorization_geturi(header)`
- `#define proxy_authorization_seturi(header, value) authorization_seturi(header, value)`
- `#define proxy_authorization_getresponse(header) authorization_getresponse(header)`
- `#define proxy_authorization_setresponse(header, value) authorization_setresponse(header, value)`
- `#define proxy_authorization_getdigest(header) authorization_getdigest(header)`
- `#define proxy_authorization_setdigest(header, value) authorization_setdigest(header, value)`
- `#define proxy_authorization_getalgorithm(header) authorization_getalgorithm(header)`
- `#define proxy_authorization_setalgorithm(header, value) authorization_setalgorithm(header, value)`
- `#define proxy_authorization_getcnonce(header) authorization_getcnonce(header)`
- `#define proxy_authorization_setcnonce(header, value) authorization_setcnonce(header, value)`
- `#define proxy_authorization_getopaque(header) authorization_getopaque(header)`
- `#define proxy_authorization_setopaque(header, value) authorization_setopaque(header, value)`
- `#define proxy_authorization_getmessage_qop(header) authorization_getmessage_qop(header)`
- `#define proxy_authorization_setmessage_qop(header, value) authorization_setmessage_qop(header, value)`

- #define **proxy\_authorization\_getnonce\_count**(header) authorization\_getnonce\_count(header)
- #define **proxy\_authorization\_setnonce\_count**(header, value) authorization\_setnonce\_count(header, value)
- #define **record\_route\_clone**(header, dest) from\_clone(header,dest)
- #define **record\_route\_seturl**(header, url) from\_seturl(**(from\_t\*)**header,url)
- #define **record\_route\_geturl**(header) from\_geturl(**(from\_t\*)**header)
- #define **record\_route\_param\_get**(header, pos, dest) from\_param\_get(**(from\_t\*)**header,pos,dest)
- #define **record\_route\_param\_add**(header, name, value) generic\_param\_add((header) → gen\_params,name,value)
- #define **record\_route\_param\_getbyname**(header, name, dest) generic\_param\_getbyname((header) → gen\_params,name,dest)
- #define **route\_clone**(header, dest) from\_clone(header,dest)
- #define **route\_seturl**(header, url) from\_seturl(**(from\_t\*)**header,url)
- #define **route\_geturl**(header) from\_geturl(**(from\_t\*)**header)
- #define **route\_param\_get**(header, pos, dest) from\_param\_get(**(from\_t\*)**header,pos,dest)
- #define **route\_param\_add**(header, name, value) generic\_param\_add((header) → gen\_params,name,value)
- #define **route\_param\_getbyname**(header, name, dest) generic\_param\_getbyname((header) → gen\_params,name,dest)
- #define **to\_setdisplayname**(header, value) from\_setdisplayname(**(from\_t\*)**header,value)
- #define **to\_getdisplayname**(header) from\_getdisplayname(**(from\_t\*)**header)
- #define **to\_seturl**(header, url) from\_seturl(**(from\_t\*)**header,url)
- #define **to\_geturl**(header) from\_geturl(**(from\_t\*)**header)
- #define **to\_param\_get**(header, pos, dest) from\_param\_get(**(from\_t\*)**header,pos,dest)
- #define **to\_param\_getbyname**(header, name, dest) generic\_param\_getbyname((header) → gen\_params,name,dest)
- #define **to\_param\_add**(header, name, value) generic\_param\_add((header) → gen\_params,name,value)
- #define **to\_set\_tag**(header, value) generic\_param\_add((header) → gen\_params, sgetcopy("tag"),value)
- #define **to\_get\_tag**(header, dest) generic\_param\_getbyname((header) → gen\_params, "tag",dest)
- #define **via\_set\_hidden**(header) generic\_param\_add((header) → via\_params,sgetcopy("hidden"),NULL)
- #define **via\_set\_ttl**(header, value) generic\_param\_add((header) → via\_params,sgetcopy("ttl"),value)
- #define **via\_set\_maddr**(header, value) generic\_param\_add((header) → via\_params,sgetcopy("maddr"),value)
- #define **via\_set\_received**(header, value) generic\_param\_add((header) → via\_params,sgetcopy("received"),value)
- #define **via\_set\_branch**(header, value) generic\_param\_add((header) → via\_params,sgetcopy("branch"),value)
- #define **via\_param\_get**(header, pos, dest) generic\_param\_get(header,pos,dest)
- #define **via\_param\_add**(header, name, value) generic\_param\_add((header) → via\_params,name,value)
- #define **via\_param\_getbyname**(header, name, dest) generic\_param\_getbyname((header) → via\_params,name,dest)
- #define **www\_authenticate\_setstale\_true**(header) www\_authenticate\_setstale(header,sgetcopy("true"))

- `#define           www_authenticate_setstale_false(header)           www_authenticate_setstale(header,sgetcopy("false"))`
- `#define           www_authenticate_setalgorithm_MD5(header)       www_authenticate_setalgorithm(header,sgetcopy("MD5"))`
- `#define msg_setdate(header, value) msg_setheader((sip_t *)header,(char *)"date",value)`
- `#define msg_getdate(header, pos, dest) msg_header_getbyname(( sip_t *)header,"date",pos,(header_t **)dest)`
- `#define msg_setencryption(header, value) msg_setheader((sip_t *)header,(char *)"encryption",value)`
- `#define msg_getencryption(header, pos, dest) msg_header_getbyname(( sip_t *)header,"encryption",pos,(header_t **)dest)`
- `#define msg_setorganization(header, value) msg_setheader((sip_t *)header,(char *)"organization",value)`
- `#define msg_getorganization(header, pos, dest) msg_header_getbyname(( sip_t *)header,"organization",pos,(header_t **)dest)`
- `#define msg_setrequire(header, value) msg_setheader((sip_t *)header,(char *)"require",value)`
- `#define msg_getrequire(header, pos, dest) msg_header_getbyname(( sip_t *)header,"require",pos,(header_t **)dest)`
- `#define msg_setsupported(header, value) msg_setheader((sip_t *)header,(char *)"supported",value)`
- `#define msg_getsupported(header, pos, dest) msg_header_getbyname(( sip_t *)header,"supported",pos,(header_t **)dest)`
- `#define msg_settimestamp(header, value) msg_setheader((sip_t *)header,(char *)"timestamp",value)`
- `#define msg_gettimestamp(header, pos, dest) msg_header_getbyname(( sip_t *)header,"timestamp",pos,(header_t **)dest)`
- `#define msg_setuser_agent(header, value) msg_setheader((sip_t *)header,(char *)"user-agent",value)`
- `#define msg_getuser_agent(header, pos, dest) msg_header_getbyname(( sip_t *)header,"user-agent",pos,(header_t **)dest)`
- `#define msg_setcontent_language(header, value) msg_setheader((sip_t *)header,(char *)"content-language",value)`
- `#define msg_getcontent_language(header, pos, dest) msg_header_getbyname(( sip_t *)header,"content-language",pos,(header_t **)dest)`
- `#define msg_setexpires(header, value) msg_setheader((sip_t *)header,(char *)"expires",value)`
- `#define msg_getexpires(header, pos, dest) msg_header_getbyname(( sip_t *)header,"expires",pos,(header_t **)dest)`
- `#define msg_setin_reply_to(header, value) msg_setheader((sip_t *)header,(char *)"in-reply-to",value)`
- `#define msg_getin_reply_to(header, pos, dest) msg_header_getbyname(( sip_t *)header,"in-reply-to",pos,(header_t **)dest)`
- `#define msg_setmax_forward(header, value) msg_setheader((sip_t *)header,(char *)"max-forward",value)`
- `#define msg_getmax_forward(header, pos, dest) msg_header_getbyname(( sip_t *)header,"max-forward",pos,(header_t **)dest)`
- `#define msg_setpriority(header, value) msg_setheader((sip_t *)header,(char *)"priority",value)`
- `#define msg_getpriority(header, pos, dest) msg_header_getbyname(( sip_t *)header,"priority",pos,(header_t **)dest)`

- `#define msg_setproxy_require(header, value) msg_setheader((sip_t *)header, (char *)"proxy-require", value)`
- `#define msg_getproxy_require(header, pos, dest) msg_header_getbyname(( sip_t *)header, "proxy-require", pos, (header_t **)dest)`
- `#define msg_setresponse_key(header, value) msg_setheader((sip_t *)header, (char *)"response-key", value)`
- `#define msg_getresponse_key(header, pos, dest) msg_header_getbyname(( sip_t *)header, "response-key", pos, (header_t **)dest)`
- `#define msg_setsubject(header, value) msg_setheader((sip_t *)header, (char *)"subject", value)`
- `#define msg_getsubject(header, pos, dest) msg_header_getbyname(( sip_t *)header, "subject", pos, (header_t **)dest)`
- `#define msg_setretry_after(header, value) msg_setheader((sip_t *)header, (char *)"retry-after", value)`
- `#define msg_getretry_after(header, pos, dest) msg_header_getbyname(( sip_t *)header, "retry-after", pos, (header_t **)dest)`
- `#define msg_setserver(header, value) msg_setheader((sip_t *)header, (char *)"server", value)`
- `#define msg_getserver(header, pos, dest) msg_header_getbyname(( sip_t *)header, "server", pos, (header_t **)dest)`
- `#define msg_setunsupported(header, value) msg_setheader((sip_t *)header, (char *)"unsupported", value)`
- `#define msg_getunsupported(header, pos, dest) msg_header_getbyname(( sip_t *)header, "unsupported", pos, (header_t **)dest)`
- `#define msg_setwarning(header, value) msg_setheader((sip_t *)header, (char *)"warning", value)`
- `#define msg_getwarning(header, pos, dest) msg_header_getbyname(( sip_t *)header, "warning", pos, (header_t **)dest)`

## Functions

- `int parser_init ()`
- `int msg_init (sip_t **sip)`
- `void msg_free (sip_t *sip)`
- `int msg_parse (sip_t *sip, char *message)`
- `int msg_2char (sip_t *sip, char **dest)`
- `int msg_clone (sip_t *sip, sip_t **dest)`
- `int msg_force_update (sip_t *sip)`
- `char * msg_getreason (int status_code)`
- `void msg_setreasonphrase (sip_t *sip, char *reason)`
- `char * msg_getreasonphrase (sip_t *sip)`
- `void msg_setstatuscode (sip_t *sip, char *statuscode)`
- `char * msg_getstatuscode (sip_t *sip)`
- `void msg_setmethod (sip_t *sip, char *method)`
- `char * msg_getmethod (sip_t *sip)`
- `void msg_setversion (sip_t *sip, char *version)`
- `char * msg_getversion (sip_t *sip)`
- `void msg_seturi (sip_t *sip, url_t *uri)`
- `url_t * msg_geturi (sip_t *sip)`
- `int msg_setaccept (sip_t *sip, char *hvalue)`
- `int msg_getaccept (sip_t *sip, int pos, accept_t **dest)`



- int **msg\_setaccept\_encoding** (sip\_t \*sip, char \*hvalue)
- int **msg\_getaccept\_encoding** (sip\_t \*sip, int pos, **accept\_encoding\_t** \*\*dest)
- int **msg\_setaccept\_language** (sip\_t \*sip, char \*hvalue)
- int **msg\_getaccept\_language** (sip\_t \*sip, int pos, **accept\_language\_t** \*\*dest)
- int **msg\_setalert\_info** (sip\_t \*sip, char \*hvalue)
- int **msg\_getalert\_info** (sip\_t \*sip, int pos, **alert\_info\_t** \*\*dest)
- int **msg\_setallow** (sip\_t \*sip, char \*hvalue)
- int **msg\_getallow** (sip\_t \*sip, int pos, **allow\_t** \*\*dest)
- int **msg\_setauthorization** (sip\_t \*sip, char \*hvalue)
- **authorization\_t** \* **msg\_getauthorization** (sip\_t \*sip)
- int **msg\_setcall\_id** (sip\_t \*sip, char \*hvalue)
- **call\_id\_t** \* **msg\_getcall\_id** (sip\_t \*sip)
- int **msg\_setcall\_info** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcall\_info** (sip\_t \*sip, int pos, **call\_info\_t** \*\*dest)
- int **msg\_setcontact** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontact** (sip\_t \*sip, int pos, **contact\_t** \*\*dest)
- int **msg\_setcontent\_disposition** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontent\_disposition** (sip\_t \*sip, int pos, **content\_disposition\_t** \*\*dest)
- int **msg\_setcontent\_encoding** (sip\_t \*sip, char \*hvalue)
- int **msg\_getcontent\_encoding** (sip\_t \*sip, int pos, **content\_encoding\_t** \*\*dest)
- int **msg\_setcontent\_length** (sip\_t \*sip, char \*hvalue)
- **content\_length\_t** \* **msg\_getcontent\_length** (sip\_t \*sip)
- int **msg\_setcontent\_type** (sip\_t \*sip, char \*hvalue)
- **content\_type\_t** \* **msg\_getcontent\_type** (sip\_t \*sip)
- int **msg\_setcseq** (sip\_t \*sip, char \*hvalue)
- **cseq\_t** \* **msg\_getcseq** (sip\_t \*sip)
- int **msg\_seterror\_info** (sip\_t \*sip, char \*hvalue)
- int **msg\_geterror\_info** (sip\_t \*sip, int pos, **error\_info\_t** \*\*dest)
- int **msg\_setfrom** (sip\_t \*sip, char \*hvalue)
- **from\_t** \* **msg\_getfrom** (sip\_t \*sip)
- int **msg\_setmime\_version** (sip\_t \*sip, char \*hvalue)
- **mime\_version\_t** \* **msg\_getmime\_version** (sip\_t \*sip)
- int **msg\_setproxy\_authenticate** (sip\_t \*sip, char \*hvalue)
- **proxy\_authenticate\_t** \* **msg\_getproxy\_authenticate** (sip\_t \*sip)
- int **msg\_setproxy\_authorization** (sip\_t \*sip, char \*hvalue)
- int **msg\_getproxy\_authorization** (sip\_t \*sip, int pos, **proxy\_authorization\_t** \*\*dest)
- int **msg\_setrecord\_route** (sip\_t \*sip, char \*hvalue)
- int **msg\_getrecord\_route** (sip\_t \*sip, int pos, **record\_route\_t** \*\*dest)
- int **msg\_setroute** (sip\_t \*sip, char \*hvalue)
- int **msg\_getroute** (sip\_t \*sip, int pos, **route\_t** \*\*dest)
- int **msg\_setto** (sip\_t \*sip, char \*hvalue)
- **to\_t** \* **msg\_getto** (sip\_t \*sip)
- int **msg\_setvia** (sip\_t \*sip, char \*hvalue)
- int **msg\_getvia** (sip\_t \*sip, int pos, **via\_t** \*\*dest)
- int **msg\_setwww\_authenticate** (sip\_t \*sip, char \*hvalue)
- **www\_authenticate\_t** \* **msg\_getwww\_authenticate** (sip\_t \*sip)
- int **msg\_setheader** (sip\_t \*sip, char \*hname, char \*hvalue)
- int **msg\_header\_getbyname** (sip\_t \*sip, char \*hname, int pos, **header\_t** \*\*dest)
- int **msg\_getheader** (sip\_t \*sip, int pos, **header\_t** \*\*dest)
- int **msg\_setbody** (sip\_t \*sip, char \*buf)

- int **msg\_setbody\_mime** (sip\_t \*sip, char \*buf)
- int **msg\_getbody** (sip\_t \*sip, int pos, body\_t \*\*dest)
- int **body\_init** (body\_t \*\*body)
- void **body\_free** (body\_t \*body)
- int **body\_parse** (body\_t \*body, char \*buf)
- int **body\_parse\_mime** (body\_t \*body, char \*buf)
- int **body\_2char** (body\_t \*body, char \*\*dest)
- void **generic\_param\_setname** (generic\_param\_t \*generic\_param, char \*name)
- char \* **generic\_param\_getname** (generic\_param\_t \*generic\_param)
- void **generic\_param\_setvalue** (generic\_param\_t \*generic\_param, char \*value)
- char \* **generic\_param\_getvalue** (generic\_param\_t \*generic\_param)
- int **header\_init** (header\_t \*\*header)
- void **header\_free** (header\_t \*header)
- int **header\_2char** (header\_t \*header, char \*\*dest)
- char \* **header\_getname** (header\_t \*header)
- void **header\_setname** (header\_t \*header, char \*pname)
- char \* **header\_getvalue** (header\_t \*header)
- void **header\_setvalue** (header\_t \*header, char \*pvalue)
- int **header\_clone** (header\_t \*header, header\_t \*\*dest)
- int **accept\_encoding\_init** (accept\_encoding\_t \*\*header)
- int **accept\_encoding\_parse** (accept\_encoding\_t \*header, char \*hvalue)
- int **accept\_encoding\_2char** (accept\_encoding\_t \*header, char \*\*dest)
- void **accept\_encoding\_free** (accept\_encoding\_t \*header)
- int **accept\_encoding\_clone** (accept\_encoding\_t \*header, accept\_encoding\_t \*\*dest)
- void **accept\_encoding\_setelement** (accept\_encoding\_t \*header, char \*value)
- char \* **accept\_encoding\_getelement** (accept\_encoding\_t \*header)
- int **authorization\_init** (authorization\_t \*\*header)
- int **authorization\_parse** (authorization\_t \*header, char \*hvalue)
- int **authorization\_2char** (authorization\_t \*header, char \*\*dest)
- void **authorization\_free** (authorization\_t \*header)
- int **authorization\_clone** (authorization\_t \*header, authorization\_t \*\*dest)
- char \* **authorization\_getauth\_type** (authorization\_t \*header)
- void **authorization\_setauth\_type** (authorization\_t \*header, char \*value)
- char \* **authorization\_getusername** (authorization\_t \*header)
- void **authorization\_setusername** (authorization\_t \*header, char \*value)
- char \* **authorization\_getrealm** (authorization\_t \*header)
- void **authorization\_setrealm** (authorization\_t \*header, char \*value)
- char \* **authorization\_getnonce** (authorization\_t \*header)
- void **authorization\_setnonce** (authorization\_t \*header, char \*value)
- char \* **authorization\_geturi** (authorization\_t \*header)
- void **authorization\_seturi** (authorization\_t \*header, char \*value)
- char \* **authorization\_getresponse** (authorization\_t \*header)
- void **authorization\_setresponse** (authorization\_t \*header, char \*value)
- char \* **authorization\_getdigest** (authorization\_t \*header)
- void **authorization\_setdigest** (authorization\_t \*header, char \*value)
- char \* **authorization\_getalgorithm** (authorization\_t \*header)
- void **authorization\_setalgorithm** (authorization\_t \*header, char \*value)
- char \* **authorization\_getcnonce** (authorization\_t \*header)
- void **authorization\_setcnonce** (authorization\_t \*header, char \*value)
- char \* **authorization\_getopaque** (authorization\_t \*header)

- void **authorization\_setopaque** (**authorization\_t** \*header, char \*value)
- char \* **authorization\_getmessage\_qop** (**authorization\_t** \*header)
- void **authorization\_setmessage\_qop** (**authorization\_t** \*header, char \*value)
- char \* **authorization\_getnonce\_count** (**authorization\_t** \*header)
- void **authorization\_setnonce\_count** (**authorization\_t** \*header, char \*value)
- int **call\_id\_init** (**call\_id\_t** \*\*header)
- void **call\_id\_free** (**call\_id\_t** \*header)
- int **call\_id\_parse** (**call\_id\_t** \*header, char \*hvalue)
- int **call\_id\_2char** (**call\_id\_t** \*header, char \*\*dest)
- int **call\_id\_clone** (**call\_id\_t** \*header, **call\_id\_t** \*\*dest)
- void **call\_id\_setnumber** (**call\_id\_t** \*header, char \*value)
- char \* **call\_id\_getnumber** (**call\_id\_t** \*header)
- void **call\_id\_sethost** (**call\_id\_t** \*header, char \*value)
- char \* **call\_id\_gethost** (**call\_id\_t** \*header)
- int **call\_info\_init** (**call\_info\_t** \*\*header)
- void **call\_info\_free** (**call\_info\_t** \*header)
- int **call\_info\_parse** (**call\_info\_t** \*header, char \*hvalue)
- int **call\_info\_2char** (**call\_info\_t** \*header, char \*\*dest)
- int **call\_info\_clone** (**call\_info\_t** \*header, **call\_info\_t** \*\*dest)
- char \* **call\_info\_geturi** (**call\_info\_t** \*header)
- void **call\_info\_seturi** (**call\_info\_t** \*header, char \*uri)
- int **contact\_init** (**contact\_t** \*\*header)
- void **contact\_free** (**contact\_t** \*header)
- int **contact\_parse** (**contact\_t** \*header, char \*hvalue)
- int **contact\_2char** (**contact\_t** \*header, char \*\*dest)
- int **contact\_clone** (**contact\_t** \*header, **contact\_t** \*\*dest)
- int **content\_disposition\_parse** (**content\_disposition\_t** \*header, char \*hvalue)
- int **content\_length\_init** (**content\_length\_t** \*\*header)
- void **content\_length\_free** (**content\_length\_t** \*header)
- int **content\_length\_parse** (**content\_length\_t** \*header, char \*hvalue)
- int **content\_length\_2char** (**content\_length\_t** \*header, char \*\*dest)
- int **content\_length\_clone** (**content\_length\_t** \*header, **content\_length\_t** \*\*dest)
- int **content\_type\_init** (**content\_type\_t** \*\*header)
- void **content\_type\_free** (**content\_type\_t** \*header)
- int **content\_type\_parse** (**content\_type\_t** \*header, char \*hvalue)
- int **content\_type\_2char** (**content\_type\_t** \*header, char \*\*dest)
- int **content\_type\_clone** (**content\_type\_t** \*header, **content\_type\_t** \*\*dest)
- int **cseq\_init** (**cseq\_t** \*\*header)
- void **cseq\_free** (**cseq\_t** \*header)
- int **cseq\_parse** (**cseq\_t** \*header, char \*hvalue)
- int **cseq\_2char** (**cseq\_t** \*header, char \*\*dest)
- int **cseq\_clone** (**cseq\_t** \*header, **cseq\_t** \*\*dest)
- void **cseq\_setnumber** (**cseq\_t** \*header, char \*value)
- char \* **cseq\_getnumber** (**cseq\_t** \*header)
- void **cseq\_setmethod** (**cseq\_t** \*header, char \*value)
- char \* **cseq\_getmethod** (**cseq\_t** \*header)
- int **from\_init** (**from\_t** \*\*header)
- void **from\_free** (**from\_t** \*header)
- int **from\_parse** (**from\_t** \*header, char \*hvalue)
- int **from\_2char** (**from\_t** \*header, char \*\*dest)

- **int from\_clone** (**from\_t** \*header, **from\_t** \*\*dest)
- **void from\_setdisplayname** (**from\_t** \*header, char \*value)
- **char \* from\_getdisplayname** (**from\_t** \*header)
- **void from\_seturl** (**from\_t** \*header, **url\_t** \*url)
- **url\_t \* from\_geturl** (**from\_t** \*header)
- **int from\_param\_get** (**from\_t** \*header, int pos, **generic\_param\_t** \*\*dest)
- **int record\_route\_init** (**record\_route\_t** \*\*header)
- **void record\_route\_free** (**record\_route\_t** \*header)
- **int record\_route\_parse** (**record\_route\_t** \*header, char \*hvalue)
- **int record\_route\_2char** (**record\_route\_t** \*header, char \*\*dest)
- **int route\_init** (**route\_t** \*\*header)
- **void route\_free** (**route\_t** \*header)
- **int route\_parse** (**route\_t** \*header, char \*hvalue)
- **int route\_2char** (**route\_t** \*header, char \*\*dest)
- **int to\_init** (**to\_t** \*\*header)
- **void to\_free** (**to\_t** \*header)
- **int to\_parse** (**to\_t** \*header, char \*hvalue)
- **int to\_2char** (**to\_t** \*header, char \*\*dest)
- **int to\_clone** (**to\_t** \*header, **to\_t** \*\*dest)
- **int via\_init** (**via\_t** \*\*header)
- **void via\_free** (**via\_t** \*header)
- **int via\_parse** (**via\_t** \*header, char \*hvalue)
- **int via\_2char** (**via\_t** \*header, char \*\*dest)
- **int via\_clone** (**via\_t** \*header, **via\_t** \*\*dest)
- **void via\_setversion** (**via\_t** \*header, char \*value)
- **char \* via\_getversion** (**via\_t** \*header)
- **void via\_setprotocol** (**via\_t** \*header, char \*value)
- **char \* via\_getprotocol** (**via\_t** \*header)
- **void via\_sethost** (**via\_t** \*header, char \*value)
- **char \* via\_gethost** (**via\_t** \*header)
- **void via\_setport** (**via\_t** \*header, char \*value)
- **char \* via\_getport** (**via\_t** \*header)
- **void via\_setcomment** (**via\_t** \*header, char \*value)
- **char \* via\_getcomment** (**via\_t** \*header)
- **int www\_authenticate\_init** (**www\_authenticate\_t** \*\*header)
- **int www\_authenticate\_parse** (**www\_authenticate\_t** \*header, char \*hvalue)
- **int www\_authenticate\_2char** (**www\_authenticate\_t** \*header, char \*\*dest)
- **void www\_authenticate\_free** (**www\_authenticate\_t** \*header)
- **int www\_authenticate\_clone** (**www\_authenticate\_t** \*header, **www\_authenticate\_t** \*\*dest)
- **char \* www\_authenticate\_getauth\_type** (**www\_authenticate\_t** \*header)
- **void www\_authenticate\_setauth\_type** (**www\_authenticate\_t** \*header, char \*value)
- **char \* www\_authenticate\_getrealm** (**www\_authenticate\_t** \*header)
- **void www\_authenticate\_setrealm** (**www\_authenticate\_t** \*header, char \*value)
- **char \* www\_authenticate\_getdomain** (**www\_authenticate\_t** \*header)
- **void www\_authenticate\_setdomain** (**www\_authenticate\_t** \*header, char \*value)
- **char \* www\_authenticate\_getnonce** (**www\_authenticate\_t** \*header)
- **void www\_authenticate\_setnonce** (**www\_authenticate\_t** \*header, char \*value)
- **char \* www\_authenticate\_getopaque** (**www\_authenticate\_t** \*header)
- **void www\_authenticate\_setopaque** (**www\_authenticate\_t** \*header, char \*value)

- char \* **www\_authenticate\_getstale** (**www\_authenticate\_t** \*header)
- void **www\_authenticate\_setstale** (**www\_authenticate\_t** \*header, char \*value)
- char \* **www\_authenticate\_getalgorithm** (**www\_authenticate\_t** \*header)
- void **www\_authenticate\_setalgorithm** (**www\_authenticate\_t** \*header, char \*value)
- char \* **www\_authenticate\_getqop\_options** (**www\_authenticate\_t** \*header)
- void **www\_authenticate\_setqop\_options** (**www\_authenticate\_t** \*header, char \*value)

### 6.8.1 Detailed Description

oSIP parser Routines.

This is the SIP accessor and parser related API.

Understanding the parser implementation will prevent you from using it improperly. Read this carefully.

This implementation could be seen as a partial implementation of the whole SIP syntax. In other words, the parser is 'tolerant' and will not detect a lot of error cases. As an example, no error will be detected while trying to parse the following request-uri:

```
INVITE sip: jack@atosc.org:abcd SIP/2.0
```

This code shows that even if your SIP message is parsed correctly by oSIP, it may still be not compliant. This could be used by attackers to make your application crash or whatever. In this example, if you are trying to call the atoi() method with the string 'abcd', your application will crash. Of course, there exist solutions! You can check yourself for the validity of the string or use the strtol() method (found on most unix) which is capable of detecting such error cases.

Are you wondering why the parser has been built this way?

The initial answer is that each SIP application have different requirement and some (the proxy!) needs SIP message to be parsed as quickly as possible. Also, most applications only need a few information from a SIP message. (the first Via is the only one interesting!). If the parser was fully checking each Via field validity, it would consume too much CPU on useless operations. If you think this model does not fit your application, then you should buy a slow stack :-).

Is there any plan to change that behaviour?

I do not need it, but if this interest you, it would be possible to compile oSIP in 2 different ways: a full checker model could be useful for SIP application with no performance requirements. Any contributions is welcomed and will be merged if it's made optional.

## 6.9 smsgtypes.h File Reference

oSIP type definitions.

```
#include <osip/list.h>
```

```
#include <osip/urls.h>
```

### Compounds

- struct **accept\_encoding\_t**
- struct **authorization\_t**
- struct **body\_t**
- struct **call\_id\_t**
- struct **call\_info\_t**
- struct **content\_length\_t**
- struct **content\_type\_t**
- struct **cseq\_t**
- struct **from\_t**
- struct **header\_t**
- struct **language\_tag\_t**
- struct **sip\_t**
- struct **startline\_t**
- struct **via\_t**
- struct **www\_authenticate\_t**

### Defines

- **#define BODY\_MESSAGE\_MAX\_SIZE** 500

### Typedefs

- typedef startline\_t **startline\_t**
- typedef header\_t **header\_t**
- typedef cseq\_t **cseq\_t**
- typedef via\_t **via\_t**
- typedef url\_param\_t **generic\_param\_t**
- typedef from\_t **from\_t**
- typedef from\_t **to\_t**
- typedef from\_t **contact\_t**
- typedef from\_t **record\_route\_t**
- typedef from\_t **route\_t**
- typedef call\_id\_t **call\_id\_t**
- typedef content\_length\_t **content\_length\_t**
- typedef language\_tag\_t **language\_tag\_t**
- typedef content\_length\_t **allow\_t**
- typedef content\_length\_t **content\_encoding\_t**
- typedef content\_length\_t **mime\_version\_t**
- typedef content\_type\_t **content\_type\_t**
- typedef content\_type\_t **accept\_t**

- typedef accept\_encoding\_t **accept\_encoding\_t**
- typedef accept\_encoding\_t **accept\_language\_t**
- typedef call\_info\_t **call\_info\_t**
- typedef call\_info\_t **alert\_info\_t**
- typedef call\_info\_t **error\_info\_t**
- typedef call\_info\_t **content\_disposition\_t**
- typedef call\_info\_t **encryption\_t**
- typedef www\_authenticate\_t **www\_authenticate\_t**
- typedef www\_authenticate\_t **proxy\_authenticate\_t**
- typedef authorization\_t **authorization\_t**
- typedef authorization\_t **proxy\_authorization\_t**
- typedef body\_t **body\_t**
- typedef sip\_t **sip\_t**

### 6.9.1 Detailed Description

oSIP type definitions.

## 6.10 thread.h File Reference

oSIP Thread Routines.

```
#include <stdio.h>
#include <errno.h>
#include <pthread.h>
```

### Typedefs

- typedef pthread\_t **sthread\_t**

### Functions

- **sthread\_t \* sthread\_create** (int stacksize, **sthread\_t** \*thread, void \*(\*func)(void \*), void \*arg)
- int **sthread\_join** (**sthread\_t** \*thread)
- int **sthread\_setpriority** (**sthread\_t** \*thread, int priority)
- void **sthread\_exit** ()

#### 6.10.1 Detailed Description

oSIP Thread Routines.

Those methods are only available if the library is compile in multi threaded mode. This is the default for oSIP.



## 6.11 urls.h File Reference

oSIP url parser Routines.

```
#include <osip/const.h>
```

### Compounds

- struct **url\_param\_t**
- struct **url\_t**

### Defines

- `#define url_header_init(url_header) url_param_init(url_header)`
- `#define url_header_free(url_header) url_param_free(url_header)`
- `#define url_header_set(url_header, name, value) url_param_set(url_header, name, value)`
- `#define url_header_clone(url_header, dest) url_param_clone(url_header, dest)`
- `#define url_header_add(url_headers, name, value) url_param_add(url_headers, name, value)`
- `#define url_header_getbyname(url_headers, name, dest) url_param_getbyname(url_headers, name, dest)`
- `#define url_set_transport_udp(url) url_param_add(url → url_params, "transport", "udp")`
- `#define url_set_transport_tcp(url) url_param_add(url → url_params, "transport", "tcp")`
- `#define url_set_transport_sctp(url) url_param_add(url → url_params, "transport", "sctp")`
- `#define url_set_transport_tls(url) url_param_add(url → url_params, "transport", "tls")`
- `#define url_set_transport(url, value) url_param_add(url → url_params, "transport", value)`
- `#define url_set_user_phone(url) url_param_add(url → url_params, "user", "phone")`
- `#define url_set_user_ip(url) url_param_add(url → url_params, "user", "ip")`
- `#define url_set_user(url, value) url_param_add(url → url_params, "user", value)`
- `#define url_set_method_invite(url) url_param_add(url → url_params, "method", "INVITE")`
- `#define url_set_method_ack(url) url_param_add(url → url_params, "method", "ACK")`
- `#define url_set_method_options(url) url_param_add(url → url_params, "method", "OPTIONS")`
- `#define url_set_method_bye(url) url_param_add(url → url_params, "method", "BYE")`
- `#define url_set_method_cancel(url) url_param_add(url → url_params, "method", "CANCEL")`
- `#define url_set_method_register(url) url_param_add(url → url_params, "method", "REGISTER")`
- `#define url_set_method(url, value) url_param_add(url → url_params, "method", value)`
- `#define url_set_ttl(url, value) url_param_add(url → url_params, "ttl", value)`
- `#define url_set_maddr(url, value) url_param_add(url → url_params, "maddr", value)`
- `#define url_uparam_get(url, pos, dest) url_param_get(url → url_params, pos, dest)`
- `#define url_uparam_add(url, name, value) url_param_add(url → url_params, name, value)`
- `#define url_uparam_getbyname(url, name, dest) url_param_getbyname(url → url_params, name, dest)`
- `#define url_uheader_get(url, pos, dest) url_header_get(url → url_headers, pos, dest)`
- `#define url_uheader_add(url, name, value) url_header_add(url → url_headers, name, value)`
- `#define url_uheader_getbyname(url, name, dest) url_header_getbyname(url → url_headers, name, dest)`

## Typedefs

- typedef url\_param\_t **url\_param\_t**
- typedef url\_param\_t **url\_header\_t**
- typedef url\_t **url\_t**

## Functions

- int **url\_param\_init** (url\_param\_t \*\*url\_param)
- void **url\_param\_free** (url\_param\_t \*url\_param)
- int **url\_param\_set** (url\_param\_t \*url\_param, char \*name, char \*value)
- int **url\_param\_clone** (url\_param\_t \*url\_param, url\_param\_t \*\*dest)
- int **url\_param\_add** (list\_t \*url\_params, char \*name, char \*value)
- int **url\_param\_getbyname** (list\_t \*url\_params, char \*name, url\_param\_t \*\*dest)
- int **url\_init** (url\_t \*\*url)
- void **url\_free** (url\_t \*url)
- int **url\_parse** (url\_t \*url, char \*buf)
- int **url\_2char** (url\_t \*url, char \*\*dest)
- int **url\_clone** (url\_t \*url, url\_t \*\*dest)
- void **url\_setscheme** (url\_t \*url, char \*value)
- char \* **url\_getscheme** (url\_t \*url)
- void **url\_sethost** (url\_t \*url, char \*value)
- char \* **url\_gethost** (url\_t \*url)
- void **url setUsername** (url\_t \*url, char \*value)
- char \* **url getUsername** (url\_t \*url)
- void **url setPassword** (url\_t \*url, char \*value)
- char \* **url getPassword** (url\_t \*url)
- void **url setport** (url\_t \*url, char \*value)
- char \* **url getport** (url\_t \*url)

### 6.11.1 Detailed Description

oSIP url parser Routines.

This is the implementation of sip url scheme. It also partially support any unrecognised scheme (not starting with 'sip:' or 'sips:'). Unrecognised scheme are stored in url->string.

---

# Index

accept\_2char  
    oSIP\_SMSG, 76

accept\_clone  
    oSIP\_SMSG, 76

accept\_encoding\_2char  
    oSIP\_SMSG, 117

accept\_encoding\_clone  
    oSIP\_SMSG, 117

accept\_encoding\_free  
    oSIP\_SMSG, 118

accept\_encoding\_getelement  
    oSIP\_SMSG, 118

accept\_encoding\_init  
    oSIP\_SMSG, 118

accept\_encoding\_param\_add  
    oSIP\_SMSG, 76

accept\_encoding\_param\_get  
    oSIP\_SMSG, 76

accept\_encoding\_param\_getbyname  
    oSIP\_SMSG, 76

accept\_encoding\_parse  
    oSIP\_SMSG, 118

accept\_encoding\_setelement  
    oSIP\_SMSG, 118

accept\_encoding\_t  
    oSIP\_TYPES, 156

accept\_free  
    oSIP\_SMSG, 76

accept\_init  
    oSIP\_SMSG, 77

accept\_language\_2char  
    oSIP\_SMSG, 77

accept\_language\_clone  
    oSIP\_SMSG, 77

accept\_language\_free  
    oSIP\_SMSG, 77

accept\_language\_getelement  
    oSIP\_SMSG, 77

accept\_language\_init  
    oSIP\_SMSG, 77

accept\_language\_param\_add  
    oSIP\_SMSG, 78

accept\_language\_param\_get  
    oSIP\_SMSG, 78

accept\_language\_param\_getbyname  
    oSIP\_SMSG, 78

accept\_language\_parse  
    oSIP\_SMSG, 78

accept\_language\_setelement  
    oSIP\_SMSG, 78

accept\_language\_t  
    oSIP\_TYPES, 156

accept\_param\_add  
    oSIP\_SMSG, 79

accept\_param\_get  
    oSIP\_SMSG, 79

accept\_param\_getbyname  
    oSIP\_SMSG, 79

accept\_parse  
    oSIP\_SMSG, 79

accept\_t  
    oSIP\_TYPES, 156

alert\_info\_2char  
    oSIP\_SMSG, 79

alert\_info\_clone  
    oSIP\_SMSG, 80

alert\_info\_free  
    oSIP\_SMSG, 80

alert\_info\_geturi  
    oSIP\_SMSG, 80

alert\_info\_init  
    oSIP\_SMSG, 80

alert\_info\_parse  
    oSIP\_SMSG, 80

alert\_info\_seturi  
    oSIP\_SMSG, 80

alert\_info\_t  
    oSIP\_TYPES, 156

allow\_2char  
    oSIP\_SMSG, 81

allow\_clone  
    oSIP\_SMSG, 81

allow\_free  
    oSIP\_SMSG, 81

allow\_init  
    oSIP\_SMSG, 81

allow\_parse  
    oSIP\_SMSG, 81

---

- allow\_t
  - oSIP\_TYPES, 156
- authorization\_2char
  - oSIP\_SMSG, 118
- authorization\_clone
  - oSIP\_SMSG, 118
- authorization\_free
  - oSIP\_SMSG, 119
- authorization\_getalgorithm
  - oSIP\_SMSG, 119
- authorization\_getauth\_type
  - oSIP\_SMSG, 119
- authorization\_getcnonce
  - oSIP\_SMSG, 119
- authorization\_getdigest
  - oSIP\_SMSG, 119
- authorization\_getmessage\_qop
  - oSIP\_SMSG, 119
- authorization\_getnonce
  - oSIP\_SMSG, 119
- authorization\_getnonce\_count
  - oSIP\_SMSG, 120
- authorization\_getopaque
  - oSIP\_SMSG, 120
- authorization\_getrealm
  - oSIP\_SMSG, 120
- authorization\_getresponse
  - oSIP\_SMSG, 120
- authorization\_geturi
  - oSIP\_SMSG, 120
- authorization\_getusername
  - oSIP\_SMSG, 120
- authorization\_init
  - oSIP\_SMSG, 120
- authorization\_parse
  - oSIP\_SMSG, 121
- authorization\_setalgorithm
  - oSIP\_SMSG, 121
- authorization\_setauth\_type
  - oSIP\_SMSG, 121
- authorization\_setcnonce
  - oSIP\_SMSG, 121
- authorization\_setdigest
  - oSIP\_SMSG, 121
- authorization\_setmessage\_qop
  - oSIP\_SMSG, 121
- authorization\_setnonce
  - oSIP\_SMSG, 122
- authorization\_setnonce\_count
  - oSIP\_SMSG, 122
- authorization\_setopaque
  - oSIP\_SMSG, 122
- authorization\_setrealm
  - oSIP\_SMSG, 122
- authorization\_setresponse
  - oSIP\_SMSG, 122
- authorization\_seturi
  - oSIP\_SMSG, 122
- authorization\_setusername
  - oSIP\_SMSG, 123
- authorization\_t
  - oSIP\_TYPES, 156
- body\_2char
  - oSIP\_SMSG, 123
- body\_free
  - oSIP\_SMSG, 123
- body\_init
  - oSIP\_SMSG, 123
- BODY\_MESSAGE\_MAX\_SIZE
  - oSIP\_TYPES, 156
- body\_parse
  - oSIP\_SMSG, 123
- body\_parse\_mime
  - oSIP\_SMSG, 123
- body\_t
  - oSIP\_TYPES, 156
- call\_id\_2char
  - oSIP\_SMSG, 124
- call\_id\_clone
  - oSIP\_SMSG, 124
- call\_id\_free
  - oSIP\_SMSG, 124
- call\_id\_gethost
  - oSIP\_SMSG, 124
- call\_id\_getnumber
  - oSIP\_SMSG, 124
- call\_id\_init
  - oSIP\_SMSG, 124
- call\_id\_parse
  - oSIP\_SMSG, 125
- call\_id\_sethost
  - oSIP\_SMSG, 125
- call\_id\_setnumber
  - oSIP\_SMSG, 125
- call\_id\_t
  - oSIP\_TYPES, 157
- call\_info\_2char
  - oSIP\_SMSG, 125
- call\_info\_clone
  - oSIP\_SMSG, 125
- call\_info\_free
  - oSIP\_SMSG, 125
- call\_info\_geturi
  - oSIP\_SMSG, 126
- call\_info\_init
  - oSIP\_SMSG, 126

- call\_info\_parse
  - oSIP\_SMSG, 126
- call\_info\_seturi
  - oSIP\_SMSG, 126
- call\_info\_t
  - oSIP\_TYPES, 157
- contact\_2char
  - oSIP\_SMSG, 126
- contact\_clone
  - oSIP\_SMSG, 126
- contact\_free
  - oSIP\_SMSG, 127
- contact\_getdisplayname
  - oSIP\_SMSG, 81
- contact\_geturl
  - oSIP\_SMSG, 82
- contact\_init
  - oSIP\_SMSG, 127
- contact\_param\_add
  - oSIP\_SMSG, 82
- contact\_param\_get
  - oSIP\_SMSG, 82
- contact\_param\_getbyname
  - oSIP\_SMSG, 82
- contact\_parse
  - oSIP\_SMSG, 127
- contact\_setdisplayname
  - oSIP\_SMSG, 82
- contact\_seturl
  - oSIP\_SMSG, 83
- contact\_t
  - oSIP\_TYPES, 157
- content\_disposition\_2char
  - oSIP\_SMSG, 83
- content\_disposition\_clone
  - oSIP\_SMSG, 83
- content\_disposition\_free
  - oSIP\_SMSG, 83
- content\_disposition\_gettype
  - oSIP\_SMSG, 83
- content\_disposition\_init
  - oSIP\_SMSG, 83
- content\_disposition\_parse
  - oSIP\_SMSG, 127
- content\_disposition\_settype
  - oSIP\_SMSG, 84
- content\_disposition\_t
  - oSIP\_TYPES, 157
- content\_encoding\_2char
  - oSIP\_SMSG, 84
- content\_encoding\_clone
  - oSIP\_SMSG, 84
- content\_encoding\_free
  - oSIP\_SMSG, 84
- content\_encoding\_init
  - oSIP\_SMSG, 84
- content\_encoding\_parse
  - oSIP\_SMSG, 84
- content\_encoding\_t
  - oSIP\_TYPES, 157
- content\_length\_2char
  - oSIP\_SMSG, 127
- content\_length\_clone
  - oSIP\_SMSG, 127
- content\_length\_free
  - oSIP\_SMSG, 128
- content\_length\_init
  - oSIP\_SMSG, 128
- content\_length\_parse
  - oSIP\_SMSG, 128
- content\_length\_t
  - oSIP\_TYPES, 157
- content\_type\_2char
  - oSIP\_SMSG, 128
- content\_type\_clone
  - oSIP\_SMSG, 128
- content\_type\_free
  - oSIP\_SMSG, 128
- content\_type\_init
  - oSIP\_SMSG, 129
- content\_type\_param\_add
  - oSIP\_SMSG, 85
- content\_type\_param\_get
  - oSIP\_SMSG, 85
- content\_type\_param\_getbyname
  - oSIP\_SMSG, 85
- content\_type\_parse
  - oSIP\_SMSG, 129
- content\_type\_t
  - oSIP\_TYPES, 157
- context\_type\_t
  - oSIP\_FSM, 19
- cseq\_2char
  - oSIP\_SMSG, 129
- cseq\_clone
  - oSIP\_SMSG, 129
- cseq\_free
  - oSIP\_SMSG, 129
- cseq\_getmethod
  - oSIP\_SMSG, 129
- cseq\_getnumber
  - oSIP\_SMSG, 130
- cseq\_init
  - oSIP\_SMSG, 130
- cseq\_parse
  - oSIP\_SMSG, 130
- cseq\_setmethod
  - oSIP\_SMSG, 130

- cseq\_setnumber
  - oSIP\_SMSG, 130
- cseq\_t
  - oSIP\_TYPES, 157
- DEFAULT\_T1
  - oSIP\_FSM, 15
- DEFAULT\_T2
  - oSIP\_FSM, 15
- DEFAULT\_T4
  - oSIP\_FSM, 15
- dialog.h, 171
- dialog\_free
  - oSIP\_DIALOG, 5
- dialog\_init\_as\_uac
  - oSIP\_DIALOG, 6
- dialog\_init\_as\_uas
  - oSIP\_DIALOG, 6
- dialog\_match\_as\_uac
  - oSIP\_DIALOG, 6
- dialog\_match\_as\_uas
  - oSIP\_DIALOG, 6
- dialog\_set\_state
  - oSIP\_DIALOG, 6
- dialog\_t
  - oSIP\_DIALOG, 5
- dialog\_update\_cseq\_as\_uas
  - oSIP\_DIALOG, 7
- dialog\_update\_route\_set\_as\_uac
  - oSIP\_DIALOG, 7
- dialog\_update\_route\_set\_as\_uas
  - oSIP\_DIALOG, 7
- dialog\_update\_tag\_as\_uac
  - oSIP\_DIALOG, 7
- encryption\_t
  - oSIP\_TYPES, 157
- error\_info\_2char
  - oSIP\_SMSG, 85
- error\_info\_clone
  - oSIP\_SMSG, 85
- error\_info\_free
  - oSIP\_SMSG, 86
- error\_info\_geturi
  - oSIP\_SMSG, 86
- error\_info\_init
  - oSIP\_SMSG, 86
- error\_info\_parse
  - oSIP\_SMSG, 86
- error\_info\_seturi
  - oSIP\_SMSG, 86
- error\_info\_t
  - oSIP\_TYPES, 157
- EVT\_IS\_INCOMINGMSG
  - oSIP\_FSM, 15
- EVT\_IS\_INCOMINGREQ
  - oSIP\_FSM, 15
- EVT\_IS\_INCOMINGRESP
  - oSIP\_FSM, 15
- EVT\_IS\_KILL\_TRANSACTION
  - oSIP\_FSM, 16
- EVT\_IS\_MSG
  - oSIP\_FSM, 16
- EVT\_IS\_OUTGOINGMSG
  - oSIP\_FSM, 16
- EVT\_IS\_OUTGOINGREQ
  - oSIP\_FSM, 16
- EVT\_IS\_OUTGOINGRESP
  - oSIP\_FSM, 17
- EVT\_IS\_RCV\_ACK
  - oSIP\_FSM, 17
- EVT\_IS\_RCV\_INVITE
  - oSIP\_FSM, 17
- EVT\_IS\_RCV\_REQUEST
  - oSIP\_FSM, 17
- EVT\_IS\_RCV\_STATUS\_1XX
  - oSIP\_FSM, 17
- EVT\_IS\_RCV\_STATUS\_2XX
  - oSIP\_FSM, 18
- EVT\_IS\_RCV\_STATUS\_3456XX
  - oSIP\_FSM, 18
- EVT\_IS\_SND\_ACK
  - oSIP\_FSM, 18
- EVT\_IS\_SND\_INVITE
  - oSIP\_FSM, 18
- EVT\_IS\_SND\_REQUEST
  - oSIP\_FSM, 18
- EVT\_IS\_SND\_STATUS\_1XX
  - oSIP\_FSM, 18
- EVT\_IS\_SND\_STATUS\_2XX
  - oSIP\_FSM, 19
- EVT\_IS\_SND\_STATUS\_3456XX
  - oSIP\_FSM, 19
- fifo.h, 172
- fifo\_add
  - oSIP\_FIFO, 8
- fifo\_free
  - oSIP\_FIFO, 8
- fifo\_get
  - oSIP\_FIFO, 8
- fifo\_init
  - oSIP\_FIFO, 8
- fifo\_t
  - oSIP\_FIFO, 8
- fifo\_tryget
  - oSIP\_FIFO, 9
- from\_2char

- oSIP\_MSG, 130
- from\_clone
  - oSIP\_MSG, 131
- from\_free
  - oSIP\_MSG, 131
- from\_get\_tag
  - oSIP\_MSG, 86
- from\_getdisplayname
  - oSIP\_MSG, 131
- from\_geturl
  - oSIP\_MSG, 131
- from\_init
  - oSIP\_MSG, 131
- from\_param\_add
  - oSIP\_MSG, 87
- from\_param\_get
  - oSIP\_MSG, 131
- from\_param\_getbyname
  - oSIP\_MSG, 87
- from\_parse
  - oSIP\_MSG, 132
- from\_set\_tag
  - oSIP\_MSG, 87
- from\_setdisplayname
  - oSIP\_MSG, 132
- from\_seturl
  - oSIP\_MSG, 132
- from\_t
  - oSIP\_TYPES, 157
- generic\_param\_add
  - oSIP\_MSG, 87
- generic\_param\_clone
  - oSIP\_MSG, 87
- generic\_param\_free
  - oSIP\_MSG, 88
- generic\_param\_getbyname
  - oSIP\_MSG, 88
- generic\_param\_getname
  - oSIP\_MSG, 132
- generic\_param\_getvalue
  - oSIP\_MSG, 132
- generic\_param\_init
  - oSIP\_MSG, 88
- generic\_param\_set
  - oSIP\_MSG, 88
- generic\_param\_setname
  - oSIP\_MSG, 132
- generic\_param\_setvalue
  - oSIP\_MSG, 133
- generic\_param\_t
  - oSIP\_TYPES, 158
- header\_2char
  - oSIP\_MSG, 133
- header\_clone
  - oSIP\_MSG, 133
- header\_free
  - oSIP\_MSG, 133
- header\_getname
  - oSIP\_MSG, 133
- header\_getvalue
  - oSIP\_MSG, 133
- header\_init
  - oSIP\_MSG, 134
- header\_setname
  - oSIP\_MSG, 134
- header\_setvalue
  - oSIP\_MSG, 134
- header\_t
  - oSIP\_TYPES, 158
- ict\_set\_destination
  - oSIP\_FSM, 20
- ict\_t
  - oSIP\_FSM, 19
- ist\_t
  - oSIP\_FSM, 19
- language\_tag\_t
  - oSIP\_TYPES, 158
- list\_h, 173
- list\_add
  - oSIP\_LIST, 10
- list\_eol
  - oSIP\_LIST, 10
- list\_get
  - oSIP\_LIST, 10
- list\_init
  - oSIP\_LIST, 11
- list\_remove
  - oSIP\_LIST, 11
- list\_size
  - oSIP\_LIST, 11
- list\_special\_free
  - oSIP\_LIST, 11
- list\_t
  - oSIP\_LIST, 10
- listofchar\_free
  - oSIP\_LIST, 11
- mime\_version\_2char
  - oSIP\_MSG, 88
- mime\_version\_clone
  - oSIP\_MSG, 88
- mime\_version\_free
  - oSIP\_MSG, 89
- mime\_version\_init

- oSIP\_SMSG, 89
- mime\_version\_parse
  - oSIP\_SMSG, 89
- mime\_version\_t
  - oSIP\_TYPES, 158
- msg\_2char
  - oSIP\_SMSG, 134
- msg\_clone
  - oSIP\_SMSG, 134
- msg\_force\_update
  - oSIP\_SMSG, 134
- msg\_free
  - oSIP\_SMSG, 135
- msg\_getaccept
  - oSIP\_SMSG, 135
- msg\_getaccept\_encoding
  - oSIP\_SMSG, 135
- msg\_getaccept\_language
  - oSIP\_SMSG, 135
- msg\_getalert\_info
  - oSIP\_SMSG, 135
- msg\_getallow
  - oSIP\_SMSG, 136
- msg\_getauthorization
  - oSIP\_SMSG, 136
- msg\_getbody
  - oSIP\_SMSG, 136
- msg\_getcall\_id
  - oSIP\_SMSG, 136
- msg\_getcall\_info
  - oSIP\_SMSG, 136
- msg\_getcontact
  - oSIP\_SMSG, 136
- msg\_getcontent\_disposition
  - oSIP\_SMSG, 137
- msg\_getcontent\_encoding
  - oSIP\_SMSG, 137
- msg\_getcontent\_language
  - oSIP\_SMSG, 89
- msg\_getcontent\_length
  - oSIP\_SMSG, 137
- msg\_getcontent\_type
  - oSIP\_SMSG, 137
- msg\_getcseq
  - oSIP\_SMSG, 137
- msg\_getdate
  - oSIP\_SMSG, 89
- msg\_getencryption
  - oSIP\_SMSG, 90
- msg\_geterror\_info
  - oSIP\_SMSG, 137
- msg\_getexpires
  - oSIP\_SMSG, 90
- msg\_getfrom
  - oSIP\_SMSG, 138
- msg\_getheader
  - oSIP\_SMSG, 138
- msg\_getin\_reply\_to
  - oSIP\_SMSG, 90
- msg\_getmax\_forward
  - oSIP\_SMSG, 90
- msg\_getmethod
  - oSIP\_SMSG, 138
- msg\_getmime\_version
  - oSIP\_SMSG, 138
- msg\_getorganization
  - oSIP\_SMSG, 90
- msg\_getpriority
  - oSIP\_SMSG, 91
- msg\_getproxy\_authenticate
  - oSIP\_SMSG, 138
- msg\_getproxy\_authorization
  - oSIP\_SMSG, 138
- msg\_getproxy\_require
  - oSIP\_SMSG, 91
- msg\_getreason
  - oSIP\_SMSG, 139
- msg\_getreasonphrase
  - oSIP\_SMSG, 139
- msg\_getrecord\_route
  - oSIP\_SMSG, 139
- msg\_getrequire
  - oSIP\_SMSG, 91
- msg\_getresponse\_key
  - oSIP\_SMSG, 91
- msg\_getretry\_after
  - oSIP\_SMSG, 91
- msg\_getroute
  - oSIP\_SMSG, 139
- msg\_getserver
  - oSIP\_SMSG, 92
- msg\_getstatuscode
  - oSIP\_SMSG, 139
- msg\_getsubject
  - oSIP\_SMSG, 92
- msg\_getsupported
  - oSIP\_SMSG, 92
- msg\_gettimestamp
  - oSIP\_SMSG, 92
- msg\_getto
  - oSIP\_SMSG, 139
- msg\_getunsupported
  - oSIP\_SMSG, 92
- msg\_geturi
  - oSIP\_SMSG, 140
- msg\_getuser\_agent
  - oSIP\_SMSG, 93
- msg\_getversion



- oSIP\_SMSG, 140
- msg\_getvia
  - oSIP\_SMSG, 140
- msg\_getwarning
  - oSIP\_SMSG, 93
- msg\_getwww\_authenticate
  - oSIP\_SMSG, 140
- msg\_header\_getbyname
  - oSIP\_SMSG, 140
- msg\_init
  - oSIP\_SMSG, 140
- MSG\_IS\_ACK
  - oSIP\_SMSG, 93
- MSG\_IS\_BYE
  - oSIP\_SMSG, 93
- MSG\_IS\_CANCEL
  - oSIP\_SMSG, 94
- MSG\_IS\_INFO
  - oSIP\_SMSG, 94
- MSG\_IS\_INVITE
  - oSIP\_SMSG, 94
- MSG\_IS\_NOTIFY
  - oSIP\_SMSG, 94
- MSG\_IS\_OPTIONS
  - oSIP\_SMSG, 94
- MSG\_IS\_PRACK
  - oSIP\_SMSG, 95
- MSG\_IS\_REGISTER
  - oSIP\_SMSG, 95
- MSG\_IS\_REQUEST
  - oSIP\_SMSG, 95
- MSG\_IS\_RESPONSE
  - oSIP\_SMSG, 95
- MSG\_IS\_RESPONSEFOR
  - oSIP\_SMSG, 95
- MSG\_IS\_STATUS\_1XX
  - oSIP\_SMSG, 96
- MSG\_IS\_STATUS\_2XX
  - oSIP\_SMSG, 96
- MSG\_IS\_STATUS\_3XX
  - oSIP\_SMSG, 96
- MSG\_IS\_STATUS\_4XX
  - oSIP\_SMSG, 96
- MSG\_IS\_STATUS\_5XX
  - oSIP\_SMSG, 97
- MSG\_IS\_STATUS\_6XX
  - oSIP\_SMSG, 97
- MSG\_IS\_SUBSCRIBE
  - oSIP\_SMSG, 97
- msg\_parse
  - oSIP\_SMSG, 141
- msg\_setaccept
  - oSIP\_SMSG, 141
- msg\_setaccept\_encoding
  - oSIP\_SMSG, 141
- msg\_setaccept\_language
  - oSIP\_SMSG, 141
- msg\_setalert\_info
  - oSIP\_SMSG, 141
- msg\_setallow
  - oSIP\_SMSG, 141
- msg\_setauthorization
  - oSIP\_SMSG, 142
- msg\_setbody
  - oSIP\_SMSG, 142
- msg\_setbody\_mime
  - oSIP\_SMSG, 142
- msg\_setcall\_id
  - oSIP\_SMSG, 142
- msg\_setcall\_info
  - oSIP\_SMSG, 142
- msg\_setcontact
  - oSIP\_SMSG, 142
- msg\_setcontent\_disposition
  - oSIP\_SMSG, 143
- msg\_setcontent\_encoding
  - oSIP\_SMSG, 143
- msg\_setcontent\_language
  - oSIP\_SMSG, 97
- msg\_setcontent\_length
  - oSIP\_SMSG, 143
- msg\_setcontent\_type
  - oSIP\_SMSG, 143
- msg\_setcseq
  - oSIP\_SMSG, 143
- msg\_setdate
  - oSIP\_SMSG, 98
- msg\_setencryption
  - oSIP\_SMSG, 98
- msg\_seterror\_info
  - oSIP\_SMSG, 143
- msg\_setexpires
  - oSIP\_SMSG, 98
- msg\_setfrom
  - oSIP\_SMSG, 144
- msg\_setheader
  - oSIP\_SMSG, 144
- msg\_setin\_reply\_to
  - oSIP\_SMSG, 98
- msg\_setmax\_forward
  - oSIP\_SMSG, 98
- msg\_setmethod
  - oSIP\_SMSG, 144
- msg\_setmime\_version
  - oSIP\_SMSG, 144
- msg\_setorganization
  - oSIP\_SMSG, 99
- msg\_setpriority

- oSIP\_SMSG, 99
- msg\_setproxy\_authenticate
  - oSIP\_SMSG, 144
- msg\_setproxy\_authorization
  - oSIP\_SMSG, 144
- msg\_setproxy\_require
  - oSIP\_SMSG, 99
- msg\_setreasonphrase
  - oSIP\_SMSG, 145
- msg\_setrecord\_route
  - oSIP\_SMSG, 145
- msg\_setrequire
  - oSIP\_SMSG, 99
- msg\_setresponse\_key
  - oSIP\_SMSG, 99
- msg\_setretry\_after
  - oSIP\_SMSG, 100
- msg\_setroute
  - oSIP\_SMSG, 145
- msg\_setserver
  - oSIP\_SMSG, 100
- msg\_setstatuscode
  - oSIP\_SMSG, 145
- msg\_setsubject
  - oSIP\_SMSG, 100
- msg\_setsupported
  - oSIP\_SMSG, 100
- msg\_settimestamp
  - oSIP\_SMSG, 100
- msg\_setto
  - oSIP\_SMSG, 145
- msg\_setunsupported
  - oSIP\_SMSG, 101
- msg\_seturi
  - oSIP\_SMSG, 145
- msg\_setuser\_agent
  - oSIP\_SMSG, 101
- msg\_setversion
  - oSIP\_SMSG, 146
- msg\_setvia
  - oSIP\_SMSG, 146
- msg\_setwarning
  - oSIP\_SMSG, 101
- msg\_setwww\_authenticate
  - oSIP\_SMSG, 146
- MSG\_TEST\_CODE
  - oSIP\_SMSG, 101
- nict\_set\_destination
  - oSIP\_FSM, 20
- nict\_t
  - oSIP\_FSM, 19
- nist\_need\_timer\_j\_event
  - oSIP\_FSM, 21
- nist\_t
  - oSIP\_FSM, 19
- oSIP and SDP offer/answer model Handling, 54
- oSIP dialog Handling, 5
- oSIP fifo Handling, 8
- oSIP fsm Handling, 12
- oSIP list Handling, 10
- oSIP parser Handling, 65
- oSIP SDP parser Handling, 39
- oSIP semaphore definitions, 63
- oSIP type definitions, 155
- oSIP url parser Handling, 160
- osip.h, 174
- osip\_create\_transaction
  - oSIP\_FSM, 21
- oSIP\_DIALOG
  - dialog\_free, 5
  - dialog\_init\_as\_uac, 6
  - dialog\_init\_as\_uas, 6
  - dialog\_match\_as\_uac, 6
  - dialog\_match\_as\_uas, 6
  - dialog\_set\_state, 6
  - dialog\_t, 5
  - dialog\_update\_cseq\_as\_uas, 7
  - dialog\_update\_route\_set\_as\_uac, 7
  - dialog\_update\_route\_set\_as\_uas, 7
  - dialog\_update\_tag\_as\_uac, 7
- oSIP\_FIFO
  - fifo\_add, 8
  - fifo\_free, 8
  - fifo\_get, 8
  - fifo\_init, 8
  - fifo\_t, 8
  - fifo\_tryget, 9
- osip\_find\_transaction
  - oSIP\_FSM, 21
- osip\_free
  - oSIP\_FSM, 21
- oSIP\_FSM
  - context\_type\_t, 19
  - DEFAULT\_T1, 15
  - DEFAULT\_T2, 15
  - DEFAULT\_T4, 15
  - EVT\_IS\_INCOMINGMSG, 15
  - EVT\_IS\_INCOMINGREQ, 15
  - EVT\_IS\_INCOMINGRESP, 15
  - EVT\_IS\_KILL\_TRANSACTION, 16
  - EVT\_IS\_MSG, 16
  - EVT\_IS\_OUTGOINGMSG, 16
  - EVT\_IS\_OUTGOINGREQ, 16
  - EVT\_IS\_OUTGOINGRESP, 17
  - EVT\_IS\_RCV\_ACK, 17

EVT\_IS\_RCV\_INVITE, 17  
EVT\_IS\_RCV\_REQUEST, 17  
EVT\_IS\_RCV\_STATUS\_1XX, 17  
EVT\_IS\_RCV\_STATUS\_2XX, 18  
EVT\_IS\_RCV\_STATUS\_3456XX, 18  
EVT\_IS\_SND\_ACK, 18  
EVT\_IS\_SND\_INVITE, 18  
EVT\_IS\_SND\_REQUEST, 18  
EVT\_IS\_SND\_STATUS\_1XX, 18  
EVT\_IS\_SND\_STATUS\_2XX, 19  
EVT\_IS\_SND\_STATUS\_3456XX, 19  
ict\_set\_destination, 20  
ict\_t, 19  
ist\_t, 19  
nict\_set\_destination, 20  
nict\_t, 19  
nist\_need\_timer\_j\_event, 21  
nist\_t, 19  
osip\_create\_transaction, 21  
osip\_find\_transaction, 21  
osip\_free, 21  
osip\_global\_free, 21  
osip\_global\_init, 21  
osip\_ict\_execute, 21  
osip\_init, 22  
osip\_ist\_execute, 22  
osip\_new\_outgoing\_sipmessage, 22  
osip\_nict\_execute, 22  
osip\_nist\_execute, 22  
osip\_parse, 22  
osip\_setcb\_ict\_1xx\_received, 22  
osip\_setcb\_ict\_2xx\_received, 23  
osip\_setcb\_ict\_2xx\_received2, 23  
osip\_setcb\_ict\_3456xx\_received2, 23  
osip\_setcb\_ict\_3xx\_received, 23  
osip\_setcb\_ict\_4xx\_received, 23  
osip\_setcb\_ict\_5xx\_received, 24  
osip\_setcb\_ict\_6xx\_received, 24  
osip\_setcb\_ict\_ack\_sent, 24  
osip\_setcb\_ict\_ack\_sent2, 24  
osip\_setcb\_ict\_invite\_sent, 24  
osip\_setcb\_ict\_invite\_sent2, 25  
osip\_setcb\_ict\_kill\_transaction, 25  
osip\_setcb\_ict\_transport\_error, 25  
osip\_setcb\_ist\_1xx\_sent, 25  
osip\_setcb\_ist\_1xx\_sent2, 25  
osip\_setcb\_ist\_2xx\_sent, 26  
osip\_setcb\_ist\_2xx\_sent2, 26  
osip\_setcb\_ist\_3456xx\_sent2, 26  
osip\_setcb\_ist\_3xx\_sent, 26  
osip\_setcb\_ist\_4xx\_sent, 26  
osip\_setcb\_ist\_5xx\_sent, 27  
osip\_setcb\_ist\_6xx\_sent, 27  
osip\_setcb\_ist\_ack\_received, 27  
osip\_setcb\_ist\_ack\_received2, 27  
osip\_setcb\_ist\_invite\_received, 27  
osip\_setcb\_ist\_invite\_received2, 28  
osip\_setcb\_ist\_kill\_transaction, 28  
osip\_setcb\_ist\_transport\_error, 28  
osip\_setcb\_nict\_1xx\_received, 28  
osip\_setcb\_nict\_2xx\_received, 28  
osip\_setcb\_nict\_2xx\_received2, 29  
osip\_setcb\_nict\_3456xx\_received2, 29  
osip\_setcb\_nict\_3xx\_received, 29  
osip\_setcb\_nict\_4xx\_received, 29  
osip\_setcb\_nict\_5xx\_received, 29  
osip\_setcb\_nict\_6xx\_received, 30  
osip\_setcb\_nict\_bye\_sent, 30  
osip\_setcb\_nict\_cancel\_sent, 30  
osip\_setcb\_nict\_info\_sent, 30  
osip\_setcb\_nict\_kill\_transaction, 30  
osip\_setcb\_nict\_notify\_sent, 31  
osip\_setcb\_nict\_options\_sent, 31  
osip\_setcb\_nict\_register\_sent, 31  
osip\_setcb\_nict\_request\_sent2, 31  
osip\_setcb\_nict\_subscribe\_sent, 31  
osip\_setcb\_nict\_transport\_error, 32  
osip\_setcb\_nict\_unknown\_sent, 32  
osip\_setcb\_nist\_1xx\_sent, 32  
osip\_setcb\_nist\_2xx\_sent, 32  
osip\_setcb\_nist\_2xx\_sent2, 32  
osip\_setcb\_nist\_3456xx\_sent2, 33  
osip\_setcb\_nist\_3xx\_sent, 33  
osip\_setcb\_nist\_4xx\_sent, 33  
osip\_setcb\_nist\_5xx\_sent, 33  
osip\_setcb\_nist\_6xx\_sent, 33  
osip\_setcb\_nist\_bye\_received, 34  
osip\_setcb\_nist\_cancel\_received, 34  
osip\_setcb\_nist\_info\_received, 34  
osip\_setcb\_nist\_kill\_transaction, 34  
osip\_setcb\_nist\_notify\_received, 34  
osip\_setcb\_nist\_options\_received, 35  
osip\_setcb\_nist\_register\_received, 35  
osip\_setcb\_nist\_request\_received2, 35  
osip\_setcb\_nist\_subscribe\_received, 35  
osip\_setcb\_nist\_transport\_error, 35  
osip\_setcb\_nist\_unknown\_received, 36  
osip\_setcb\_send\_message, 36  
osip\_t, 20  
osip\_timers\_ict\_execute, 36  
osip\_timers\_ist\_execute, 36  
osip\_timers\_nict\_execute, 36  
osip\_timers\_nist\_execute, 36  
osip\_transaction\_find, 37  
SIP\_MESSAGE\_MAX\_LENGTH, 19  
sipevent\_t, 20  
transaction\_execute, 37  
transaction\_free, 37

- transaction\_get\_your\_instance, 37
- transaction\_init, 37
- transaction\_set\_your\_instance, 37
- transaction\_t, 20
- type\_t, 20
- osip\_global\_free
  - oSIP\_FSM, 21
- osip\_global\_init
  - oSIP\_FSM, 21
- osip\_ict\_execute
  - oSIP\_FSM, 21
- osip\_init
  - oSIP\_FSM, 22
- osip\_ist\_execute
  - oSIP\_FSM, 22
- oSIP\_LIST
  - list\_add, 10
  - list\_eol, 10
  - list\_get, 10
  - list\_init, 11
  - list\_remove, 11
  - list\_size, 11
  - list\_special\_free, 11
  - list\_t, 10
  - listofchar\_free, 11
- osip\_new\_outgoing\_sipmessage
  - oSIP\_FSM, 22
- osip\_nict\_execute
  - oSIP\_FSM, 22
- osip\_nist\_execute
  - oSIP\_FSM, 22
- oSIP\_OAM
  - payload\_free, 55
  - payload\_init, 55
  - payload\_t, 55
  - sdp\_config\_add\_support\_for\_audio\_codec, 56
  - sdp\_config\_add\_support\_for\_other\_codec, 56
  - sdp\_config\_add\_support\_for\_video\_codec, 56
  - sdp\_config\_free, 57
  - sdp\_config\_init, 57
  - sdp\_config\_set\_c\_addr, 57
  - sdp\_config\_set\_c\_addr\_multicast\_int, 57
  - sdp\_config\_set\_c\_addr\_multicast\_ttl, 57
  - sdp\_config\_set\_c\_addrtype, 57
  - sdp\_config\_set\_c\_nettype, 58
  - sdp\_config\_set\_fcn\_accept\_audio\_codec, 58
  - sdp\_config\_set\_fcn\_accept\_other\_codec, 58
  - sdp\_config\_set\_fcn\_accept\_video\_codec, 58
  - sdp\_config\_set\_fcn\_get\_audio\_port, 58
  - sdp\_config\_set\_fcn\_get\_other\_port, 58
  - sdp\_config\_set\_fcn\_get\_video\_port, 59
  - sdp\_config\_set\_fcn\_set\_attributes, 59
  - sdp\_config\_set\_fcn\_set\_emails, 59
  - sdp\_config\_set\_fcn\_set\_info, 59
  - sdp\_config\_set\_fcn\_set\_phones, 59
  - sdp\_config\_set\_fcn\_set\_uri, 59
  - sdp\_config\_set\_o\_addr, 60
  - sdp\_config\_set\_o\_addrtype, 60
  - sdp\_config\_set\_o\_nettype, 60
  - sdp\_config\_set\_o\_session\_id, 60
  - sdp\_config\_set\_o\_session\_version, 60
  - sdp\_config\_set\_o\_username, 60
  - sdp\_config\_t, 55
  - sdp\_context\_execute\_negotiation, 61
  - sdp\_context\_free, 61
  - sdp\_context\_get\_local\_sdp, 61
  - sdp\_context\_get\_mycontext, 61
  - sdp\_context\_get\_remote\_sdp, 61
  - sdp\_context\_init, 61
  - sdp\_context\_set\_local\_sdp, 62
  - sdp\_context\_set\_mycontext, 62
  - sdp\_context\_set\_remote\_sdp, 62
  - sdp\_context\_t, 55
- osip\_parse
  - oSIP\_FSM, 22
- oSIP\_SDP
  - sdp\_2char, 41
  - sdp\_a\_att\_field\_get, 41
  - sdp\_a\_att\_value\_get, 41
  - sdp\_a\_attribute\_add, 42
  - sdp\_attribute\_free, 42
  - sdp\_attribute\_get, 42
  - sdp\_attribute\_init, 42
  - sdp\_attribute\_t, 40
  - sdp\_b\_bandwidth\_add, 42
  - sdp\_b\_bandwidth\_get, 43
  - sdp\_b\_bwtype\_get, 43
  - sdp\_bandwidth\_free, 43
  - sdp\_bandwidth\_get, 43
  - sdp\_bandwidth\_init, 43
  - sdp\_bandwidth\_t, 40
  - sdp\_c\_addr\_get, 44
  - sdp\_c\_addr\_multicast\_int\_get, 44
  - sdp\_c\_addr\_multicast\_ttl\_get, 44
  - sdp\_c\_addrtype\_get, 44
  - sdp\_c\_connection\_add, 44
  - sdp\_c\_nettype\_get, 45
  - sdp\_connection\_free, 45
  - sdp\_connection\_init, 45
  - sdp\_connection\_t, 41
  - sdp\_e\_email\_add, 45
  - sdp\_e\_email\_get, 45

- sdp\_endof\_media, 46
- sdp\_free, 46
- sdp\_i\_info\_get, 46
- sdp\_i\_info\_set, 46
- sdp\_init, 46
- sdp\_k\_key\_set, 46
- sdp\_k\_keydata\_get, 47
- sdp\_k\_keytype\_get, 47
- sdp\_key\_free, 47
- sdp\_key\_init, 47
- sdp\_key\_t, 41
- sdp\_m\_media\_add, 47
- sdp\_m\_media\_get, 48
- sdp\_m\_number\_of\_port\_get, 48
- sdp\_m\_payload\_add, 48
- sdp\_m\_payload\_get, 48
- sdp\_m\_port\_get, 48
- sdp\_m\_proto\_get, 48
- sdp\_media\_free, 49
- sdp\_media\_init, 49
- sdp\_media\_t, 41
- sdp\_o\_addr\_get, 49
- sdp\_o\_addrtype\_get, 49
- sdp\_o\_nettype\_get, 49
- sdp\_o\_origin\_set, 49
- sdp\_o\_sess\_id\_get, 50
- sdp\_o\_sess\_version\_get, 50
- sdp\_o\_username\_get, 50
- sdp\_p\_phone\_add, 50
- sdp\_p\_phone\_get, 50
- sdp\_parse, 51
- sdp\_r\_repeat\_add, 51
- sdp\_r\_repeat\_get, 51
- sdp\_s\_name\_get, 51
- sdp\_s\_name\_set, 51
- sdp\_t, 41
- sdp\_t\_start\_time\_get, 51
- sdp\_t\_stop\_time\_get, 52
- sdp\_t\_time\_descr\_add, 52
- sdp\_time\_descr\_free, 52
- sdp\_time\_descr\_init, 52
- sdp\_time\_descr\_t, 41
- sdp\_u\_uri\_get, 52
- sdp\_u\_uri\_set, 52
- sdp\_v\_version\_get, 53
- sdp\_v\_version\_set, 53
- sdp\_z\_adjustments\_get, 53
- sdp\_z\_adjustments\_set, 53
- oSIP\_SEMA
  - smutex\_destroy, 63
  - smutex\_init, 63
  - smutex\_lock, 63
  - smutex\_t, 63
  - smutex\_unlock, 63
  - ssem\_destroy, 64
  - ssem\_init, 64
  - ssem\_post, 64
  - ssem\_t, 63
  - ssem\_trywait, 64
  - ssem\_wait, 64
- osip\_setcb\_ict\_1xx\_received
  - oSIP\_FSM, 22
- osip\_setcb\_ict\_2xx\_received
  - oSIP\_FSM, 23
- osip\_setcb\_ict\_2xx\_received2
  - oSIP\_FSM, 23
- osip\_setcb\_ict\_3456xx\_received2
  - oSIP\_FSM, 23
- osip\_setcb\_ict\_3xx\_received
  - oSIP\_FSM, 23
- osip\_setcb\_ict\_4xx\_received
  - oSIP\_FSM, 23
- osip\_setcb\_ict\_5xx\_received
  - oSIP\_FSM, 24
- osip\_setcb\_ict\_6xx\_received
  - oSIP\_FSM, 24
- osip\_setcb\_ict\_ack\_sent
  - oSIP\_FSM, 24
- osip\_setcb\_ict\_ack\_sent2
  - oSIP\_FSM, 24
- osip\_setcb\_ict\_invite\_sent
  - oSIP\_FSM, 24
- osip\_setcb\_ict\_invite\_sent2
  - oSIP\_FSM, 25
- osip\_setcb\_ict\_kill\_transaction
  - oSIP\_FSM, 25
- osip\_setcb\_ict\_transport\_error
  - oSIP\_FSM, 25
- osip\_setcb\_ist\_1xx\_sent
  - oSIP\_FSM, 25
- osip\_setcb\_ist\_1xx\_sent2
  - oSIP\_FSM, 25
- osip\_setcb\_ist\_2xx\_sent
  - oSIP\_FSM, 26
- osip\_setcb\_ist\_2xx\_sent2
  - oSIP\_FSM, 26
- osip\_setcb\_ist\_3456xx\_sent2
  - oSIP\_FSM, 26
- osip\_setcb\_ist\_3xx\_sent
  - oSIP\_FSM, 26
- osip\_setcb\_ist\_4xx\_sent
  - oSIP\_FSM, 26
- osip\_setcb\_ist\_5xx\_sent
  - oSIP\_FSM, 27
- osip\_setcb\_ist\_6xx\_sent
  - oSIP\_FSM, 27
- osip\_setcb\_ist\_ack\_received
  - oSIP\_FSM, 27

- osip\_setcb\_ist\_ack\_received2
  - oSIP\_FSM, 27
- osip\_setcb\_ist\_invite\_received
  - oSIP\_FSM, 27
- osip\_setcb\_ist\_invite\_received2
  - oSIP\_FSM, 28
- osip\_setcb\_ist\_kill\_transaction
  - oSIP\_FSM, 28
- osip\_setcb\_ist\_transport\_error
  - oSIP\_FSM, 28
- osip\_setcb\_nict\_1xx\_received
  - oSIP\_FSM, 28
- osip\_setcb\_nict\_2xx\_received
  - oSIP\_FSM, 28
- osip\_setcb\_nict\_2xx\_received2
  - oSIP\_FSM, 29
- osip\_setcb\_nict\_3456xx\_received2
  - oSIP\_FSM, 29
- osip\_setcb\_nict\_3xx\_received
  - oSIP\_FSM, 29
- osip\_setcb\_nict\_4xx\_received
  - oSIP\_FSM, 29
- osip\_setcb\_nict\_5xx\_received
  - oSIP\_FSM, 29
- osip\_setcb\_nict\_6xx\_received
  - oSIP\_FSM, 30
- osip\_setcb\_nict\_bye\_sent
  - oSIP\_FSM, 30
- osip\_setcb\_nict\_cancel\_sent
  - oSIP\_FSM, 30
- osip\_setcb\_nict\_info\_sent
  - oSIP\_FSM, 30
- osip\_setcb\_nict\_kill\_transaction
  - oSIP\_FSM, 30
- osip\_setcb\_nict\_notify\_sent
  - oSIP\_FSM, 31
- osip\_setcb\_nict\_options\_sent
  - oSIP\_FSM, 31
- osip\_setcb\_nict\_register\_sent
  - oSIP\_FSM, 31
- osip\_setcb\_nict\_request\_sent2
  - oSIP\_FSM, 31
- osip\_setcb\_nict\_subscribe\_sent
  - oSIP\_FSM, 31
- osip\_setcb\_nict\_transport\_error
  - oSIP\_FSM, 32
- osip\_setcb\_nict\_unknown\_sent
  - oSIP\_FSM, 32
- osip\_setcb\_nist\_1xx\_sent
  - oSIP\_FSM, 32
- osip\_setcb\_nist\_2xx\_sent
  - oSIP\_FSM, 32
- osip\_setcb\_nist\_2xx\_sent2
  - oSIP\_FSM, 32
- osip\_setcb\_nist\_3456xx\_sent2
  - oSIP\_FSM, 33
- osip\_setcb\_nist\_3xx\_sent
  - oSIP\_FSM, 33
- osip\_setcb\_nist\_4xx\_sent
  - oSIP\_FSM, 33
- osip\_setcb\_nist\_5xx\_sent
  - oSIP\_FSM, 33
- osip\_setcb\_nist\_6xx\_sent
  - oSIP\_FSM, 33
- osip\_setcb\_nist\_bye\_received
  - oSIP\_FSM, 34
- osip\_setcb\_nist\_cancel\_received
  - oSIP\_FSM, 34
- osip\_setcb\_nist\_info\_received
  - oSIP\_FSM, 34
- osip\_setcb\_nist\_kill\_transaction
  - oSIP\_FSM, 34
- osip\_setcb\_nist\_notify\_received
  - oSIP\_FSM, 34
- osip\_setcb\_nist\_options\_received
  - oSIP\_FSM, 35
- osip\_setcb\_nist\_register\_received
  - oSIP\_FSM, 35
- osip\_setcb\_nist\_request\_received2
  - oSIP\_FSM, 35
- osip\_setcb\_nist\_subscribe\_received
  - oSIP\_FSM, 35
- osip\_setcb\_nist\_transport\_error
  - oSIP\_FSM, 35
- osip\_setcb\_nist\_unknown\_received
  - oSIP\_FSM, 36
- osip\_setcb\_send\_message
  - oSIP\_FSM, 36
- oSIP\_SMSG
  - accept\_2char, 76
  - accept\_clone, 76
  - accept\_encoding\_2char, 117
  - accept\_encoding\_clone, 117
  - accept\_encoding\_free, 118
  - accept\_encoding\_getelement, 118
  - accept\_encoding\_init, 118
  - accept\_encoding\_param\_add, 76
  - accept\_encoding\_param\_get, 76
  - accept\_encoding\_param\_getbyname, 76
  - accept\_encoding\_parse, 118
  - accept\_encoding\_setelement, 118
  - accept\_free, 76
  - accept\_init, 77
  - accept\_language\_2char, 77
  - accept\_language\_clone, 77
  - accept\_language\_free, 77
  - accept\_language\_getelement, 77
  - accept\_language\_init, 77

accept\_language\_param\_add, 78  
accept\_language\_param\_get, 78  
accept\_language\_param\_getbyname, 78  
accept\_language\_parse, 78  
accept\_language\_setelement, 78  
accept\_param\_add, 79  
accept\_param\_get, 79  
accept\_param\_getbyname, 79  
accept\_parse, 79  
alert\_info\_2char, 79  
alert\_info\_clone, 80  
alert\_info\_free, 80  
alert\_info\_geturi, 80  
alert\_info\_init, 80  
alert\_info\_parse, 80  
alert\_info\_seturi, 80  
allow\_2char, 81  
allow\_clone, 81  
allow\_free, 81  
allow\_init, 81  
allow\_parse, 81  
authorization\_2char, 118  
authorization\_clone, 118  
authorization\_free, 119  
authorization\_getalgorithm, 119  
authorization\_getauth\_type, 119  
authorization\_getcnonce, 119  
authorization\_getdigest, 119  
authorization\_getmessage\_qop, 119  
authorization\_getnonce, 119  
authorization\_getnonce\_count, 120  
authorization\_getopaque, 120  
authorization\_getrealm, 120  
authorization\_getresponse, 120  
authorization\_geturi, 120  
authorization\_getusername, 120  
authorization\_init, 120  
authorization\_parse, 121  
authorization\_setalgorithm, 121  
authorization\_setauth\_type, 121  
authorization\_setcnonce, 121  
authorization\_setdigest, 121  
authorization\_setmessage\_qop, 121  
authorization\_setnonce, 122  
authorization\_setnonce\_count, 122  
authorization\_setopaque, 122  
authorization\_setrealm, 122  
authorization\_setresponse, 122  
authorization\_seturi, 122  
authorization\_setusername, 123  
body\_2char, 123  
body\_free, 123  
body\_init, 123  
body\_parse, 123  
body\_parse\_mime, 123  
call\_id\_2char, 124  
call\_id\_clone, 124  
call\_id\_free, 124  
call\_id\_gethost, 124  
call\_id\_getnumber, 124  
call\_id\_init, 124  
call\_id\_parse, 125  
call\_id\_sethost, 125  
call\_id\_setnumber, 125  
call\_info\_2char, 125  
call\_info\_clone, 125  
call\_info\_free, 125  
call\_info\_geturi, 126  
call\_info\_init, 126  
call\_info\_parse, 126  
call\_info\_seturi, 126  
contact\_2char, 126  
contact\_clone, 126  
contact\_free, 127  
contact\_getdisplayname, 81  
contact\_geturl, 82  
contact\_init, 127  
contact\_param\_add, 82  
contact\_param\_get, 82  
contact\_param\_getbyname, 82  
contact\_parse, 127  
contact\_setdisplayname, 82  
contact\_seturl, 83  
content\_disposition\_2char, 83  
content\_disposition\_clone, 83  
content\_disposition\_free, 83  
content\_disposition\_gettype, 83  
content\_disposition\_init, 83  
content\_disposition\_parse, 127  
content\_disposition\_settype, 84  
content\_encoding\_2char, 84  
content\_encoding\_clone, 84  
content\_encoding\_free, 84  
content\_encoding\_init, 84  
content\_encoding\_parse, 84  
content\_length\_2char, 127  
content\_length\_clone, 127  
content\_length\_free, 128  
content\_length\_init, 128  
content\_length\_parse, 128  
content\_type\_2char, 128  
content\_type\_clone, 128  
content\_type\_free, 128  
content\_type\_init, 129  
content\_type\_param\_add, 85  
content\_type\_param\_get, 85  
content\_type\_param\_getbyname, 85  
content\_type\_parse, 129

cseq\_2char, 129  
cseq\_clone, 129  
cseq\_free, 129  
cseq\_getmethod, 129  
cseq\_getnumber, 130  
cseq\_init, 130  
cseq\_parse, 130  
cseq\_setmethod, 130  
cseq\_setnumber, 130  
error\_info\_2char, 85  
error\_info\_clone, 85  
error\_info\_free, 86  
error\_info\_geturi, 86  
error\_info\_init, 86  
error\_info\_parse, 86  
error\_info\_seturi, 86  
from\_2char, 130  
from\_clone, 131  
from\_free, 131  
from\_get\_tag, 86  
from\_getdisplayname, 131  
from\_geturl, 131  
from\_init, 131  
from\_param\_add, 87  
from\_param\_get, 131  
from\_param\_getbyname, 87  
from\_parse, 132  
from\_set\_tag, 87  
from\_setdisplayname, 132  
from\_seturl, 132  
generic\_param\_add, 87  
generic\_param\_clone, 87  
generic\_param\_free, 88  
generic\_param\_getbyname, 88  
generic\_param\_getname, 132  
generic\_param\_getvalue, 132  
generic\_param\_init, 88  
generic\_param\_set, 88  
generic\_param\_setname, 132  
generic\_param\_setvalue, 133  
header\_2char, 133  
header\_clone, 133  
header\_free, 133  
header\_getname, 133  
header\_getvalue, 133  
header\_init, 134  
header\_setname, 134  
header\_setvalue, 134  
mime\_version\_2char, 88  
mime\_version\_clone, 88  
mime\_version\_free, 89  
mime\_version\_init, 89  
mime\_version\_parse, 89  
msg\_2char, 134  
msg\_clone, 134  
msg\_force\_update, 134  
msg\_free, 135  
msg\_getaccept, 135  
msg\_getaccept\_encoding, 135  
msg\_getaccept\_language, 135  
msg\_getalert\_info, 135  
msg\_getallow, 136  
msg\_getauthorization, 136  
msg\_getbody, 136  
msg\_getcall\_id, 136  
msg\_getcall\_info, 136  
msg\_getcontact, 136  
msg\_getcontent\_disposition, 137  
msg\_getcontent\_encoding, 137  
msg\_getcontent\_language, 89  
msg\_getcontent\_length, 137  
msg\_getcontent\_type, 137  
msg\_getcseq, 137  
msg\_getdate, 89  
msg\_getencryption, 90  
msg\_geterror\_info, 137  
msg\_getexpires, 90  
msg\_getfrom, 138  
msg\_getheader, 138  
msg\_getin\_reply\_to, 90  
msg\_getmax\_forward, 90  
msg\_getmethod, 138  
msg\_getmime\_version, 138  
msg\_getorganization, 90  
msg\_getpriority, 91  
msg\_getproxy\_authenticate, 138  
msg\_getproxy\_authorization, 138  
msg\_getproxy\_require, 91  
msg\_getreason, 139  
msg\_getreasonphrase, 139  
msg\_getrecord\_route, 139  
msg\_getrequire, 91  
msg\_getresponse\_key, 91  
msg\_getretry\_after, 91  
msg\_getroute, 139  
msg\_getserver, 92  
msg\_getstatusCode, 139  
msg\_getsubject, 92  
msg\_getsupported, 92  
msg\_gettimestamp, 92  
msg\_getto, 139  
msg\_getunsupported, 92  
msg\_geturi, 140  
msg\_getuser\_agent, 93  
msg\_getversion, 140  
msg\_getvia, 140  
msg\_getwarning, 93  
msg\_getwww\_authenticate, 140



msg\_header\_getbyname, 140  
msg\_init, 140  
MSG\_IS\_ACK, 93  
MSG\_IS\_BYE, 93  
MSG\_IS\_CANCEL, 94  
MSG\_IS\_INFO, 94  
MSG\_IS\_INVITE, 94  
MSG\_IS\_NOTIFY, 94  
MSG\_IS\_OPTIONS, 94  
MSG\_IS\_PRACK, 95  
MSG\_IS\_REGISTER, 95  
MSG\_IS\_REQUEST, 95  
MSG\_IS\_RESPONSE, 95  
MSG\_IS\_RESPONSEFOR, 95  
MSG\_IS\_STATUS\_1XX, 96  
MSG\_IS\_STATUS\_2XX, 96  
MSG\_IS\_STATUS\_3XX, 96  
MSG\_IS\_STATUS\_4XX, 96  
MSG\_IS\_STATUS\_5XX, 97  
MSG\_IS\_STATUS\_6XX, 97  
MSG\_IS\_SUBSCRIBE, 97  
msg\_parse, 141  
msg\_setaccept, 141  
msg\_setaccept\_encoding, 141  
msg\_setaccept\_language, 141  
msg\_setalert\_info, 141  
msg\_setallow, 141  
msg\_setauthorization, 142  
msg\_setbody, 142  
msg\_setbody\_mime, 142  
msg\_setcall\_id, 142  
msg\_setcall\_info, 142  
msg\_setcontact, 142  
msg\_setcontent\_disposition, 143  
msg\_setcontent\_encoding, 143  
msg\_setcontent\_language, 97  
msg\_setcontent\_length, 143  
msg\_setcontent\_type, 143  
msg\_setcseq, 143  
msg\_setdate, 98  
msg\_setencryption, 98  
msg\_seterror\_info, 143  
msg\_setexpires, 98  
msg\_setfrom, 144  
msg\_setheader, 144  
msg\_setin\_reply\_to, 98  
msg\_setmax\_forward, 98  
msg\_setmethod, 144  
msg\_setmime\_version, 144  
msg\_setorganization, 99  
msg\_setpriority, 99  
msg\_setproxy\_authenticate, 144  
msg\_setproxy\_authorization, 144  
msg\_setproxy\_require, 99  
msg\_setreasonphrase, 145  
msg\_setrecord\_route, 145  
msg\_setrequire, 99  
msg\_setresponse\_key, 99  
msg\_setretry\_after, 100  
msg\_setroute, 145  
msg\_setserver, 100  
msg\_setstatuscode, 145  
msg\_setsubject, 100  
msg\_setsupported, 100  
msg\_settimestamp, 100  
msg\_setto, 145  
msg\_setunsupported, 101  
msg\_seturi, 145  
msg\_setuser\_agent, 101  
msg\_setversion, 146  
msg\_setvia, 146  
msg\_setwarning, 101  
msg\_setwww\_authenticate, 146  
MSG\_TEST\_CODE, 101  
parser\_init, 146  
proxy\_authenticate\_2char, 101  
proxy\_authenticate\_clone, 102  
proxy\_authenticate\_free, 102  
proxy\_authenticate\_getalgorithm, 102  
proxy\_authenticate\_getauth\_type, 102  
proxy\_authenticate\_getdomain, 102  
proxy\_authenticate\_getnonce, 102  
proxy\_authenticate\_getopaque, 103  
proxy\_authenticate\_getqop\_options, 103  
proxy\_authenticate\_getrealm, 103  
proxy\_authenticate\_getstale, 103  
proxy\_authenticate\_init, 103  
proxy\_authenticate\_parse, 103  
proxy\_authenticate\_setalgorithm, 104  
proxy\_authenticate\_setalgorithm\_MD5, 104  
proxy\_authenticate\_setauth\_type, 104  
proxy\_authenticate\_setdomain, 104  
proxy\_authenticate\_setnonce, 104  
proxy\_authenticate\_setopaque, 105  
proxy\_authenticate\_setqop\_options, 105  
proxy\_authenticate\_setrealm, 105  
proxy\_authenticate\_setstale, 105  
proxy\_authenticate\_setstale\_false, 105  
proxy\_authenticate\_setstale\_true, 106  
proxy\_authorization\_2char, 106  
proxy\_authorization\_clone, 106  
proxy\_authorization\_free, 106  
proxy\_authorization\_getalgorithm, 106  
proxy\_authorization\_getauth\_type, 106  
proxy\_authorization\_getcnonce, 107  
proxy\_authorization\_getdigest, 107

- proxy\_authorization\_getmessage\_qop, 107
- proxy\_authorization\_getnonce, 107
- proxy\_authorization\_getnonce\_count, 107
- proxy\_authorization\_getopaque, 107
- proxy\_authorization\_getrealm, 108
- proxy\_authorization\_getresponse, 108
- proxy\_authorization\_geturi, 108
- proxy\_authorization\_getusername, 108
- proxy\_authorization\_init, 108
- proxy\_authorization\_parse, 108
- proxy\_authorization\_setalgorithm, 109
- proxy\_authorization\_setauth\_type, 109
- proxy\_authorization\_setnonce, 109
- proxy\_authorization\_setdigest, 109
- proxy\_authorization\_setmessage\_qop, 109
- proxy\_authorization\_setnonce, 110
- proxy\_authorization\_setnonce\_count, 110
- proxy\_authorization\_setopaque, 110
- proxy\_authorization\_setrealm, 110
- proxy\_authorization\_setresponse, 110
- proxy\_authorization\_seturi, 111
- proxy\_authorization\_setusername, 111
- record\_route\_2char, 146
- record\_route\_clone, 111
- record\_route\_free, 146
- record\_route\_geturl, 111
- record\_route\_init, 147
- record\_route\_param\_add, 111
- record\_route\_param\_get, 112
- record\_route\_param\_getbyname, 112
- record\_route\_parse, 147
- record\_route\_seturl, 112
- route\_2char, 147
- route\_clone, 112
- route\_free, 147
- route\_geturl, 112
- route\_init, 147
- route\_param\_add, 112
- route\_param\_get, 113
- route\_param\_getbyname, 113
- route\_parse, 147
- route\_seturl, 113
- to\_2char, 148
- to\_clone, 148
- to\_free, 148
- to\_get\_tag, 113
- to\_getdisplayname, 113
- to\_geturl, 114
- to\_init, 148
- to\_param\_add, 114
- to\_param\_get, 114
- to\_param\_getbyname, 114
- to\_parse, 148
- to\_set\_tag, 114
- to\_setdisplayname, 115
- to\_seturl, 115
- via\_2char, 148
- via\_clone, 149
- via\_free, 149
- via\_getcomment, 149
- via\_gethost, 149
- via\_getport, 149
- via\_getprotocol, 149
- via\_getversion, 149
- via\_init, 150
- via\_param\_add, 115
- via\_param\_get, 115
- via\_param\_getbyname, 115
- via\_parse, 150
- via\_set\_branch, 116
- via\_set\_hidden, 116
- via\_set\_maddr, 116
- via\_set\_received, 116
- via\_set\_ttl, 116
- via\_setcomment, 150
- via\_sethost, 150
- via\_setport, 150
- via\_setprotocol, 150
- via\_setversion, 151
- www\_authenticate\_2char, 151
- www\_authenticate\_clone, 151
- www\_authenticate\_free, 151
- www\_authenticate\_getalgorithm, 151
- www\_authenticate\_getauth\_type, 151
- www\_authenticate\_getdomain, 152
- www\_authenticate\_getnonce, 152
- www\_authenticate\_getopaque, 152
- www\_authenticate\_getqop\_options, 152
- www\_authenticate\_getrealm, 152
- www\_authenticate\_getstale, 152
- www\_authenticate\_init, 152
- www\_authenticate\_parse, 153
- www\_authenticate\_setalgorithm, 153
- www\_authenticate\_setalgorithm\_MD5, 117
- www\_authenticate\_setauth\_type, 153
- www\_authenticate\_setdomain, 153
- www\_authenticate\_setnonce, 153
- www\_authenticate\_setopaque, 153
- www\_authenticate\_setqop\_options, 154
- www\_authenticate\_setrealm, 154
- www\_authenticate\_setstale, 154
- www\_authenticate\_setstale\_false, 117
- www\_authenticate\_setstale\_true, 117

- osip\_t
  - oSIP\_FSM, 20
- osip\_timers\_ict\_execute
  - oSIP\_FSM, 36
- osip\_timers\_ist\_execute
  - oSIP\_FSM, 36
- osip\_timers\_nict\_execute
  - oSIP\_FSM, 36
- osip\_timers\_nist\_execute
  - oSIP\_FSM, 36
- osip\_transaction\_find
  - oSIP\_FSM, 37
- oSIP\_TYPES
  - accept\_encoding\_t, 156
  - accept\_language\_t, 156
  - accept\_t, 156
  - alert\_info\_t, 156
  - allow\_t, 156
  - authorization\_t, 156
  - BODY\_MESSAGE\_MAX\_SIZE, 156
  - body\_t, 156
  - call\_id\_t, 157
  - call\_info\_t, 157
  - contact\_t, 157
  - content\_disposition\_t, 157
  - content\_encoding\_t, 157
  - content\_length\_t, 157
  - content\_type\_t, 157
  - cseq\_t, 157
  - encryption\_t, 157
  - error\_info\_t, 157
  - from\_t, 157
  - generic\_param\_t, 158
  - header\_t, 158
  - language\_tag\_t, 158
  - mime\_version\_t, 158
  - proxy\_authenticate\_t, 158
  - proxy\_authorization\_t, 158
  - record\_route\_t, 158
  - route\_t, 158
  - SIP\_MESSAGE\_MAX\_LENGTH, 156
  - sip\_t, 158
  - startline\_t, 158
  - to\_t, 158
  - via\_t, 159
  - www\_authenticate\_t, 159
- oSIP\_URLS
  - url\_2char, 167
  - url\_clone, 167
  - url\_free, 167
  - url\_gethost, 167
  - url\_getpassword, 167
  - url\_getport, 168
  - url\_getscheme, 168
  - url\_getusername, 168
  - url\_header\_add, 161
  - url\_header\_clone, 161
  - url\_header\_free, 161
  - url\_header\_getbyname, 162
  - url\_header\_init, 162
  - url\_header\_set, 162
  - url\_header\_t, 167
  - url\_init, 168
  - url\_param\_add, 168
  - url\_param\_clone, 168
  - url\_param\_free, 169
  - url\_param\_getbyname, 169
  - url\_param\_init, 169
  - url\_param\_set, 169
  - url\_param\_t, 167
  - url\_parse, 169
  - url\_set\_maddr, 162
  - url\_set\_method, 162
  - url\_set\_method\_ack, 163
  - url\_set\_method\_bye, 163
  - url\_set\_method\_cancel, 163
  - url\_set\_method\_invite, 163
  - url\_set\_method\_options, 163
  - url\_set\_method\_register, 163
  - url\_set\_transport, 164
  - url\_set\_transport\_sctp, 164
  - url\_set\_transport\_tcp, 164
  - url\_set\_transport\_tls, 164
  - url\_set\_transport\_udp, 164
  - url\_set\_ttl, 164
  - url\_set\_user, 165
  - url\_set\_user\_ip, 165
  - url\_set\_user\_phone, 165
  - url\_sethost, 169
  - url\_setpassword, 170
  - url\_setport, 170
  - url\_setscheme, 170
  - url\_setusername, 170
  - url\_t, 167
  - url\_uheader\_add, 165
  - url\_uheader\_get, 165
  - url\_uheader\_getbyname, 166
  - url\_uparam\_add, 166
  - url\_uparam\_get, 166
  - url\_uparam\_getbyname, 166
- parser\_init
  - oSIP\_MSG, 146
- payload\_free
  - oSIP\_OAM, 55
- payload\_init
  - oSIP\_OAM, 55
- payload\_t

- oSIP\_OAM, 55
- proxy\_authenticate\_2char
  - oSIP\_SMSG, 101
- proxy\_authenticate\_clone
  - oSIP\_SMSG, 102
- proxy\_authenticate\_free
  - oSIP\_SMSG, 102
- proxy\_authenticate\_getalgorithm
  - oSIP\_SMSG, 102
- proxy\_authenticate\_getauth\_type
  - oSIP\_SMSG, 102
- proxy\_authenticate\_getdomain
  - oSIP\_SMSG, 102
- proxy\_authenticate\_getnonce
  - oSIP\_SMSG, 102
- proxy\_authenticate\_getopaque
  - oSIP\_SMSG, 103
- proxy\_authenticate\_getqop\_options
  - oSIP\_SMSG, 103
- proxy\_authenticate\_getrealm
  - oSIP\_SMSG, 103
- proxy\_authenticate\_getstale
  - oSIP\_SMSG, 103
- proxy\_authenticate\_init
  - oSIP\_SMSG, 103
- proxy\_authenticate\_parse
  - oSIP\_SMSG, 103
- proxy\_authenticate\_setalgorithm
  - oSIP\_SMSG, 104
- proxy\_authenticate\_setalgorithm\_MD5
  - oSIP\_SMSG, 104
- proxy\_authenticate\_setauth\_type
  - oSIP\_SMSG, 104
- proxy\_authenticate\_setdomain
  - oSIP\_SMSG, 104
- proxy\_authenticate\_setnonce
  - oSIP\_SMSG, 104
- proxy\_authenticate\_setopaque
  - oSIP\_SMSG, 105
- proxy\_authenticate\_setqop\_options
  - oSIP\_SMSG, 105
- proxy\_authenticate\_setrealm
  - oSIP\_SMSG, 105
- proxy\_authenticate\_setstale
  - oSIP\_SMSG, 105
- proxy\_authenticate\_setstale\_false
  - oSIP\_SMSG, 105
- proxy\_authenticate\_setstale\_true
  - oSIP\_SMSG, 106
- proxy\_authenticate\_t
  - oSIP\_TYPES, 158
- proxy\_authorization\_2char
  - oSIP\_SMSG, 106
- proxy\_authorization\_clone
  - oSIP\_SMSG, 106
- proxy\_authorization\_free
  - oSIP\_SMSG, 106
- proxy\_authorization\_getalgorithm
  - oSIP\_SMSG, 106
- proxy\_authorization\_getauth\_type
  - oSIP\_SMSG, 106
- proxy\_authorization\_getnonce
  - oSIP\_SMSG, 107
- proxy\_authorization\_getdigest
  - oSIP\_SMSG, 107
- proxy\_authorization\_getmessage\_qop
  - oSIP\_SMSG, 107
- proxy\_authorization\_getnonce
  - oSIP\_SMSG, 107
- proxy\_authorization\_getnonce\_count
  - oSIP\_SMSG, 107
- proxy\_authorization\_getopaque
  - oSIP\_SMSG, 107
- proxy\_authorization\_getrealm
  - oSIP\_SMSG, 108
- proxy\_authorization\_getresponse
  - oSIP\_SMSG, 108
- proxy\_authorization\_geturi
  - oSIP\_SMSG, 108
- proxy\_authorization\_getusername
  - oSIP\_SMSG, 108
- proxy\_authorization\_init
  - oSIP\_SMSG, 108
- proxy\_authorization\_parse
  - oSIP\_SMSG, 108
- proxy\_authorization\_setalgorithm
  - oSIP\_SMSG, 109
- proxy\_authorization\_setauth\_type
  - oSIP\_SMSG, 109
- proxy\_authorization\_setnonce
  - oSIP\_SMSG, 109
- proxy\_authorization\_setdigest
  - oSIP\_SMSG, 109
- proxy\_authorization\_setmessage\_qop
  - oSIP\_SMSG, 109
- proxy\_authorization\_setnonce
  - oSIP\_SMSG, 110
- proxy\_authorization\_setnonce\_count
  - oSIP\_SMSG, 110
- proxy\_authorization\_setopaque
  - oSIP\_SMSG, 110
- proxy\_authorization\_setrealm
  - oSIP\_SMSG, 110
- proxy\_authorization\_setresponse
  - oSIP\_SMSG, 110
- proxy\_authorization\_seturi
  - oSIP\_SMSG, 111
- proxy\_authorization\_setusername

- oSIP\_SMSG, 111
- proxy\_authorization\_t
  - oSIP\_TYPES, 158
- record\_route\_2char
  - oSIP\_SMSG, 146
- record\_route\_clone
  - oSIP\_SMSG, 111
- record\_route\_free
  - oSIP\_SMSG, 146
- record\_route\_geturl
  - oSIP\_SMSG, 111
- record\_route\_init
  - oSIP\_SMSG, 147
- record\_route\_param\_add
  - oSIP\_SMSG, 111
- record\_route\_param\_get
  - oSIP\_SMSG, 112
- record\_route\_param\_getbyname
  - oSIP\_SMSG, 112
- record\_route\_parse
  - oSIP\_SMSG, 147
- record\_route\_seturl
  - oSIP\_SMSG, 112
- record\_route\_t
  - oSIP\_TYPES, 158
- route\_2char
  - oSIP\_SMSG, 147
- route\_clone
  - oSIP\_SMSG, 112
- route\_free
  - oSIP\_SMSG, 147
- route\_geturl
  - oSIP\_SMSG, 112
- route\_init
  - oSIP\_SMSG, 147
- route\_param\_add
  - oSIP\_SMSG, 112
- route\_param\_get
  - oSIP\_SMSG, 113
- route\_param\_getbyname
  - oSIP\_SMSG, 113
- route\_parse
  - oSIP\_SMSG, 147
- route\_seturl
  - oSIP\_SMSG, 113
- route\_t
  - oSIP\_TYPES, 158
- sdp.h, 178
- sdp\_2char
  - oSIP\_SDP, 41
- sdp\_a\_att\_field\_get
  - oSIP\_SDP, 41
- sdp\_a\_att\_value\_get
  - oSIP\_SDP, 41
- sdp\_a\_attribute\_add
  - oSIP\_SDP, 42
- sdp\_attribute\_free
  - oSIP\_SDP, 42
- sdp\_attribute\_get
  - oSIP\_SDP, 42
- sdp\_attribute\_init
  - oSIP\_SDP, 42
- sdp\_attribute\_t
  - oSIP\_SDP, 40
- sdp\_b\_bandwidth\_add
  - oSIP\_SDP, 42
- sdp\_b\_bandwidth\_get
  - oSIP\_SDP, 43
- sdp\_b\_bwtype\_get
  - oSIP\_SDP, 43
- sdp\_bandwidth\_free
  - oSIP\_SDP, 43
- sdp\_bandwidth\_get
  - oSIP\_SDP, 43
- sdp\_bandwidth\_init
  - oSIP\_SDP, 43
- sdp\_bandwidth\_t
  - oSIP\_SDP, 40
- sdp\_c\_addr\_get
  - oSIP\_SDP, 44
- sdp\_c\_addr\_multicast\_int\_get
  - oSIP\_SDP, 44
- sdp\_c\_addr\_multicast\_ttl\_get
  - oSIP\_SDP, 44
- sdp\_c\_addrtype\_get
  - oSIP\_SDP, 44
- sdp\_c\_connection\_add
  - oSIP\_SDP, 44
- sdp\_c\_nettype\_get
  - oSIP\_SDP, 45
- sdp\_config\_add\_support\_for\_audio\_codec
  - oSIP\_OAM, 56
- sdp\_config\_add\_support\_for\_other\_codec
  - oSIP\_OAM, 56
- sdp\_config\_add\_support\_for\_video\_codec
  - oSIP\_OAM, 56
- sdp\_config\_free
  - oSIP\_OAM, 57
- sdp\_config\_init
  - oSIP\_OAM, 57
- sdp\_config\_set\_c\_addr
  - oSIP\_OAM, 57
- sdp\_config\_set\_c\_addr\_multicast\_int
  - oSIP\_OAM, 57
- sdp\_config\_set\_c\_addr\_multicast\_ttl
  - oSIP\_OAM, 57

- sdp\_config\_set\_c\_addrtype
  - oSIP\_OAM, 57
- sdp\_config\_set\_c\_nettype
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_accept\_audio\_codec
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_accept\_other\_codec
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_accept\_video\_codec
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_get\_audio\_port
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_get\_other\_port
  - oSIP\_OAM, 58
- sdp\_config\_set\_fcn\_get\_video\_port
  - oSIP\_OAM, 59
- sdp\_config\_set\_fcn\_set\_attributes
  - oSIP\_OAM, 59
- sdp\_config\_set\_fcn\_set\_emails
  - oSIP\_OAM, 59
- sdp\_config\_set\_fcn\_set\_info
  - oSIP\_OAM, 59
- sdp\_config\_set\_fcn\_set\_phones
  - oSIP\_OAM, 59
- sdp\_config\_set\_fcn\_set\_uri
  - oSIP\_OAM, 59
- sdp\_config\_set\_o\_addr
  - oSIP\_OAM, 60
- sdp\_config\_set\_o\_addrtype
  - oSIP\_OAM, 60
- sdp\_config\_set\_o\_nettype
  - oSIP\_OAM, 60
- sdp\_config\_set\_o\_session\_id
  - oSIP\_OAM, 60
- sdp\_config\_set\_o\_session\_version
  - oSIP\_OAM, 60
- sdp\_config\_set\_o\_username
  - oSIP\_OAM, 60
- sdp\_config\_t
  - oSIP\_OAM, 55
- sdp\_connection\_free
  - oSIP\_SDP, 45
- sdp\_connection\_init
  - oSIP\_SDP, 45
- sdp\_connection\_t
  - oSIP\_SDP, 41
- sdp\_context\_execute\_negotiation
  - oSIP\_OAM, 61
- sdp\_context\_free
  - oSIP\_OAM, 61
- sdp\_context\_get\_local\_sdp
  - oSIP\_OAM, 61
- sdp\_context\_get\_mycontext
  - oSIP\_OAM, 61
- sdp\_context\_get\_remote\_sdp
  - oSIP\_OAM, 61
- sdp\_context\_init
  - oSIP\_OAM, 61
- sdp\_context\_set\_local\_sdp
  - oSIP\_OAM, 62
- sdp\_context\_set\_mycontext
  - oSIP\_OAM, 62
- sdp\_context\_set\_remote\_sdp
  - oSIP\_OAM, 62
- sdp\_context\_t
  - oSIP\_OAM, 55
- sdp\_e\_email\_add
  - oSIP\_SDP, 45
- sdp\_e\_email\_get
  - oSIP\_SDP, 45
- sdp\_endof\_media
  - oSIP\_SDP, 46
- sdp\_free
  - oSIP\_SDP, 46
- sdp\_i\_info\_get
  - oSIP\_SDP, 46
- sdp\_i\_info\_set
  - oSIP\_SDP, 46
- sdp\_init
  - oSIP\_SDP, 46
- sdp\_k\_key\_set
  - oSIP\_SDP, 46
- sdp\_k\_keydata\_get
  - oSIP\_SDP, 47
- sdp\_k\_keytype\_get
  - oSIP\_SDP, 47
- sdp\_key\_free
  - oSIP\_SDP, 47
- sdp\_key\_init
  - oSIP\_SDP, 47
- sdp\_key\_t
  - oSIP\_SDP, 41
- sdp\_m\_media\_add
  - oSIP\_SDP, 47
- sdp\_m\_media\_get
  - oSIP\_SDP, 48
- sdp\_m\_number\_of\_port\_get
  - oSIP\_SDP, 48
- sdp\_m\_payload\_add
  - oSIP\_SDP, 48
- sdp\_m\_payload\_get
  - oSIP\_SDP, 48
- sdp\_m\_port\_get
  - oSIP\_SDP, 48
- sdp\_m\_proto\_get
  - oSIP\_SDP, 48
- sdp\_media\_free
  - oSIP\_SDP, 49

- sdp\_media\_init
  - oSIP\_SDP, 49
- sdp\_media\_t
  - oSIP\_SDP, 41
- sdp\_negoc.h, 180
- sdp\_o\_addr\_get
  - oSIP\_SDP, 49
- sdp\_o\_addrtype\_get
  - oSIP\_SDP, 49
- sdp\_o\_nettype\_get
  - oSIP\_SDP, 49
- sdp\_o\_origin\_set
  - oSIP\_SDP, 49
- sdp\_o\_sess\_id\_get
  - oSIP\_SDP, 50
- sdp\_o\_sess\_version\_get
  - oSIP\_SDP, 50
- sdp\_o\_username\_get
  - oSIP\_SDP, 50
- sdp\_p\_phone\_add
  - oSIP\_SDP, 50
- sdp\_p\_phone\_get
  - oSIP\_SDP, 50
- sdp\_parse
  - oSIP\_SDP, 51
- sdp\_r\_repeat\_add
  - oSIP\_SDP, 51
- sdp\_r\_repeat\_get
  - oSIP\_SDP, 51
- sdp\_s\_name\_get
  - oSIP\_SDP, 51
- sdp\_s\_name\_set
  - oSIP\_SDP, 51
- sdp\_t
  - oSIP\_SDP, 41
- sdp\_t\_start\_time\_get
  - oSIP\_SDP, 51
- sdp\_t\_stop\_time\_get
  - oSIP\_SDP, 52
- sdp\_t\_time\_descr\_add
  - oSIP\_SDP, 52
- sdp\_time\_descr\_free
  - oSIP\_SDP, 52
- sdp\_time\_descr\_init
  - oSIP\_SDP, 52
- sdp\_time\_descr\_t
  - oSIP\_SDP, 41
- sdp\_u\_uri\_get
  - oSIP\_SDP, 52
- sdp\_u\_uri\_set
  - oSIP\_SDP, 52
- sdp\_v\_version\_get
  - oSIP\_SDP, 53
- sdp\_v\_version\_set
  - oSIP\_SDP, 53
- sdp\_z\_adjustments\_get
  - oSIP\_SDP, 53
- sdp\_z\_adjustments\_set
  - oSIP\_SDP, 53
- sema.h, 182
- SIP\_MESSAGE\_MAX\_LENGTH
  - oSIP\_FSM, 19
  - oSIP\_TYPES, 156
- sip\_t
  - oSIP\_TYPES, 158
- sipevent\_t
  - oSIP\_FSM, 20
- smsg.h, 183
- smsgtypes.h, 195
- smutex\_destroy
  - oSIP\_SEMA, 63
- smutex\_init
  - oSIP\_SEMA, 63
- smutex\_lock
  - oSIP\_SEMA, 63
- smutex\_t
  - oSIP\_SEMA, 63
- smutex\_unlock
  - oSIP\_SEMA, 63
- ssem\_destroy
  - oSIP\_SEMA, 64
- ssem\_init
  - oSIP\_SEMA, 64
- ssem\_post
  - oSIP\_SEMA, 64
- ssem\_t
  - oSIP\_SEMA, 63
- ssem\_trywait
  - oSIP\_SEMA, 64
- ssem\_wait
  - oSIP\_SEMA, 64
- startline\_t
  - oSIP\_TYPES, 158
- to\_2char
  - oSIP\_SMSG, 148
- to\_clone
  - oSIP\_SMSG, 148
- to\_free
  - oSIP\_SMSG, 148
- to\_get\_tag
  - oSIP\_SMSG, 113
- to\_getdisplayname
  - oSIP\_SMSG, 113
- to\_geturl
  - oSIP\_SMSG, 114
- to\_init
  - oSIP\_SMSG, 148

- to\_param\_add
  - oSIP\_SMSG, 114
- to\_param\_get
  - oSIP\_SMSG, 114
- to\_param\_getbyname
  - oSIP\_SMSG, 114
- to\_parse
  - oSIP\_SMSG, 148
- to\_set\_tag
  - oSIP\_SMSG, 114
- to\_setdisplayname
  - oSIP\_SMSG, 115
- to\_seturl
  - oSIP\_SMSG, 115
- to\_t
  - oSIP\_TYPES, 158
- transaction\_execute
  - oSIP\_FSM, 37
- transaction\_free
  - oSIP\_FSM, 37
- transaction\_get\_your\_instance
  - oSIP\_FSM, 37
- transaction\_init
  - oSIP\_FSM, 37
- transaction\_set\_your\_instance
  - oSIP\_FSM, 37
- transaction\_t
  - oSIP\_FSM, 20
- type\_t
  - oSIP\_FSM, 20
- url\_2char
  - oSIP\_URLS, 167
- url\_clone
  - oSIP\_URLS, 167
- url\_free
  - oSIP\_URLS, 167
- url\_gethost
  - oSIP\_URLS, 167
- url\_getpassword
  - oSIP\_URLS, 167
- url\_getport
  - oSIP\_URLS, 168
- url\_getscheme
  - oSIP\_URLS, 168
- url\_getusername
  - oSIP\_URLS, 168
- url\_header\_add
  - oSIP\_URLS, 161
- url\_header\_clone
  - oSIP\_URLS, 161
- url\_header\_free
  - oSIP\_URLS, 161
- url\_header\_getbyname
  - oSIP\_URLS, 162
- url\_header\_init
  - oSIP\_URLS, 162
- url\_header\_set
  - oSIP\_URLS, 162
- url\_header\_t
  - oSIP\_URLS, 167
- url\_init
  - oSIP\_URLS, 168
- url\_param\_add
  - oSIP\_URLS, 168
- url\_param\_clone
  - oSIP\_URLS, 168
- url\_param\_free
  - oSIP\_URLS, 169
- url\_param\_getbyname
  - oSIP\_URLS, 169
- url\_param\_init
  - oSIP\_URLS, 169
- url\_param\_set
  - oSIP\_URLS, 169
- url\_param\_t
  - oSIP\_URLS, 167
- url\_parse
  - oSIP\_URLS, 169
- url\_set\_maddr
  - oSIP\_URLS, 162
- url\_set\_method
  - oSIP\_URLS, 162
- url\_set\_method\_ack
  - oSIP\_URLS, 163
- url\_set\_method\_bye
  - oSIP\_URLS, 163
- url\_set\_method\_cancel
  - oSIP\_URLS, 163
- url\_set\_method\_invite
  - oSIP\_URLS, 163
- url\_set\_method\_options
  - oSIP\_URLS, 163
- url\_set\_method\_register
  - oSIP\_URLS, 163
- url\_set\_transport
  - oSIP\_URLS, 164
- url\_set\_transport\_sctp
  - oSIP\_URLS, 164
- url\_set\_transport\_tcp
  - oSIP\_URLS, 164
- url\_set\_transport\_tls
  - oSIP\_URLS, 164
- url\_set\_transport\_udp
  - oSIP\_URLS, 164
- url\_set\_ttl
  - oSIP\_URLS, 164
- url\_set\_user



- oSIP\_URLS, 165
- url\_set\_user\_ip
  - oSIP\_URLS, 165
- url\_set\_user\_phone
  - oSIP\_URLS, 165
- url\_sethost
  - oSIP\_URLS, 169
- url\_setpassword
  - oSIP\_URLS, 170
- url\_setport
  - oSIP\_URLS, 170
- url\_setscheme
  - oSIP\_URLS, 170
- url\_setusername
  - oSIP\_URLS, 170
- url\_t
  - oSIP\_URLS, 167
- url\_uheader\_add
  - oSIP\_URLS, 165
- url\_uheader\_get
  - oSIP\_URLS, 165
- url\_uheader\_getbyname
  - oSIP\_URLS, 166
- url\_uparam\_add
  - oSIP\_URLS, 166
- url\_uparam\_get
  - oSIP\_URLS, 166
- url\_uparam\_getbyname
  - oSIP\_URLS, 166
- urls.h, 197
- via\_2char
  - oSIP\_SMSG, 148
- via\_clone
  - oSIP\_SMSG, 149
- via\_free
  - oSIP\_SMSG, 149
- via\_getcomment
  - oSIP\_SMSG, 149
- via\_gethost
  - oSIP\_SMSG, 149
- via\_getport
  - oSIP\_SMSG, 149
- via\_getprotocol
  - oSIP\_SMSG, 149
- via\_getversion
  - oSIP\_SMSG, 149
- via\_init
  - oSIP\_SMSG, 150
- via\_param\_add
  - oSIP\_SMSG, 115
- via\_param\_get
  - oSIP\_SMSG, 115
- via\_param\_getbyname
  - oSIP\_SMSG, 115
- via\_parse
  - oSIP\_SMSG, 150
- via\_set\_branch
  - oSIP\_SMSG, 116
- via\_set\_hidden
  - oSIP\_SMSG, 116
- via\_set\_maddr
  - oSIP\_SMSG, 116
- via\_set\_received
  - oSIP\_SMSG, 116
- via\_set\_ttl
  - oSIP\_SMSG, 116
- via\_setcomment
  - oSIP\_SMSG, 150
- via\_sethost
  - oSIP\_SMSG, 150
- via\_setport
  - oSIP\_SMSG, 150
- via\_setprotocol
  - oSIP\_SMSG, 150
- via\_setversion
  - oSIP\_SMSG, 151
- via\_t
  - oSIP\_TYPES, 159
- www\_authenticate\_2char
  - oSIP\_SMSG, 151
- www\_authenticate\_clone
  - oSIP\_SMSG, 151
- www\_authenticate\_free
  - oSIP\_SMSG, 151
- www\_authenticate\_getalgorithm
  - oSIP\_SMSG, 151
- www\_authenticate\_getauth\_type
  - oSIP\_SMSG, 151
- www\_authenticate\_getdomain
  - oSIP\_SMSG, 152
- www\_authenticate\_getnonce
  - oSIP\_SMSG, 152
- www\_authenticate\_getopaque
  - oSIP\_SMSG, 152
- www\_authenticate\_getqop\_options
  - oSIP\_SMSG, 152
- www\_authenticate\_getrealm
  - oSIP\_SMSG, 152
- www\_authenticate\_getstale
  - oSIP\_SMSG, 152
- www\_authenticate\_init
  - oSIP\_SMSG, 152
- www\_authenticate\_parse
  - oSIP\_SMSG, 153
- www\_authenticate\_setalgorithm
  - oSIP\_SMSG, 153

- www\_authenticate\_setalgorithm\_MD5
  - oSIP\_SMSG, 117
- www\_authenticate\_setauth\_type
  - oSIP\_SMSG, 153
- www\_authenticate\_setdomain
  - oSIP\_SMSG, 153
- www\_authenticate\_setnonce
  - oSIP\_SMSG, 153
- www\_authenticate\_setopaque
  - oSIP\_SMSG, 153
- www\_authenticate\_setqop\_options
  - oSIP\_SMSG, 154
- www\_authenticate\_setrealm
  - oSIP\_SMSG, 154
- www\_authenticate\_setstale
  - oSIP\_SMSG, 154
- www\_authenticate\_setstale\_false
  - oSIP\_SMSG, 117
- www\_authenticate\_setstale\_true
  - oSIP\_SMSG, 117
- www\_authenticate\_t
  - oSIP\_TYPES, 159