

Bitlynx Technologies Inc.

## **STUN library**

Technical Reference Manual

# STUN library Data Structure Index

## STUN library Data Structures

Here are the data structures with brief descriptions:

<b>StunAddress4</b> .....	4
<b>StunAttrAddress4</b> .....	5
<b>StunAttrChangeRequest</b> .....	6
<b>StunAttrError</b> .....	7
<b>StunAttrHdr</b> .....	8
<b>StunAttrIntegrity</b> .....	9
<b>StunAttrString</b> .....	10
<b>StunAttrUnknown</b> .....	11
<b>StunMessage</b> .....	12
<b>StunMsgHdr</b> .....	15

# STUN library File Index

## STUN library File List

Here is a list of all files with brief descriptions:

<b>stun.h</b> .....	16
<b>stunio.h</b> .....	21

# STUN library Data Structure Documentation

## StunAddress4 Struct Reference

StunAddress4

---

### Detailed Description

IPv4 address.

### Data Fields

- `uint16_t port`
  - `uint32_t addr`
- 

### Field Documentation

`uint16_t port`

`uint32_t addr`

## StunAttrAddress4 Struct Reference

StunAttrAddress4

---

### Detailed Description

Format of IPv4 address inside STUN packet.

### Data Fields

- `uint8_t pad`
  - `uint8_t family`
  - `StunAddress4 ipv4`
- 

### Field Documentation

`uint8_t pad`

`uint8_t family`

`StunAddress4 ipv4`

## StunAttrChangeRequest Struct Reference

StunAttrChangeRequest

---

### Detailed Description

STUN attribute to change server port and/or address.

### Data Fields

- uint32\_t value
- 

### Field Documentation

**uint32\_t value**

## StunAttrError Struct Reference

StunAttrError

---

### Detailed Description

STUN attribute to indicate error.

### Data Fields

- `uint16_t pad`
  - `uint8_t error_class`
  - `uint8_t number`
  - `char reason [STUN_MAX_STRING]`
  - `uint16_t reason_length`
- 

### Field Documentation

`uint16_t pad`

`uint8_t error_class`

`uint8_t number`

`char reason[STUN_MAX_STRING]`

`uint16_t reason_length`

## StunAttrHdr Struct Reference

StunAttrHdr

---

### Detailed Description

Attribute header for each STUN attribute.

### Data Fields

- uint16\_t **type**
  - uint16\_t **length**
- 

### Field Documentation

**uint16\_t type**

**uint16\_t length**

## StunAttrIntegrity Struct Reference

StunAttrIntegrity

---

### Detailed Description

STUN attribute to specify message integrity using a SHA-1 HMAC.

### Data Fields

- char **hash** [20]
- 

### Field Documentation

**char hash[20]**

## StunAttrString Struct Reference

StunAttrString

---

### Detailed Description

STUN attribute to indicate a generic string.

### Data Fields

- char **value** [STUN\_MAX\_STRING]
  - uint16\_t **length**
- 

### Field Documentation

**char value**[STUN\_MAX\_STRING]

**uint16\_t length**

## StunAttrUnknown Struct Reference

StunAttrUnknown

---

### Detailed Description

STUN attribute to indicate unsupported attributes.

### Data Fields

- `uint16_t type` [STUN\_MAX\_UNKNOWN\_ATTRIBUTES]
  - `uint16_t length`
- 

### Field Documentation

`uint16_t type`[STUN\_MAX\_UNKNOWN\_ATTRIBUTES]

`uint16_t length`

# StunMessage Struct Reference

StunMessage

---

## Detailed Description

STUN Message.

## Data Fields

- **StunMsgHdr** hdr
  - int **has\_MappedAddress**
  - **StunAttrAddress4** mappedAddress
  - int **has\_ResponseAddress**
  - **StunAttrAddress4** responseAddress
  - int **has\_ChangeRequest**
  - **StunAttrChangeRequest** changeRequest
  - int **has\_SourceAddress**
  - **StunAttrAddress4** sourceAddress
  - int **has\_ChangedAddress**
  - **StunAttrAddress4** changedAddress
  - int **has\_Username**
  - **StunAttrString** username
  - int **has\_Password**
  - **StunAttrString** password
  - int **has\_MessageIntegrity**
  - **StunAttrIntegrity** messageIntegrity
  - int **has\_ErrorCode**
  - **StunAttrError** errorCode
  - int **has\_UnknownAttributes**
  - **StunAttrUnknown** unknownAttributes
  - int **has\_ReflectedFrom**
  - **StunAttrAddress4** reflectedFrom
  - int **has\_XorMappedAddress**
  - **StunAttrAddress4** xorMappedAddress
  - int **xorOnly**
  - int **has\_ServerName**
  - **StunAttrString** serverName
  - int **has\_SecondaryAddress**
  - **StunAttrAddress4** secondaryAddress
-

## Field Documentation

**StunMsgHdr** hdr

int has\_MappedAddress

**StunAttrAddress4** mappedAddress

int has\_ResponseAddress

**StunAttrAddress4** responseAddress

int has\_ChangeRequest

**StunAttrChangeRequest** changeRequest

int has\_SourceAddress

**StunAttrAddress4** sourceAddress

int has\_ChangedAddress

**StunAttrAddress4** changedAddress

int has\_Username

**StunAttrString** username

int has\_Password

**StunAttrString** password

int has\_MessageIntegrity

**StunAttrIntegrity** messageIntegrity

int has\_ErrorCode

**StunAttrError** errorCode

int has\_UnknownAttributes

**StunAttrUnknown** unknownAttributes

int has\_ReflectedFrom

**StunAttrAddress4** reflectedFrom

**int has\_XorMappedAddress**

**StunAttrAddress4 xorMappedAddress**

**int xorOnly**

**int has\_ServerName**

**StunAttrString serverName**

**int has\_SecondaryAddress**

**StunAttrAddress4 secondaryAddress**

## StunMsgHdr Struct Reference

StunMsgHdr

---

### Detailed Description

Packet header for each STUN packet.

### Data Fields

- uint16\_t **type**
- uint16\_t **length**
- uint8\_t **id** [16]

---

### Field Documentation

**uint16\_t type**

**uint16\_t length**

**uint8\_t id[16]**

# STUN library File Documentation

## stun.h File Reference

stun.h

---

### Detailed Description

**stun.h** specifies the basic interface for parsing STUN packets and generating STUN messages.

### Data Structures

- struct **StunMsgHdr**
- struct **StunAttrHdr**
- struct **StunAddress4**
- struct **StunAttrAddress4**
- struct **StunAttrChangeRequest**
- struct **StunAttrError**
- struct **StunAttrUnknown**
- struct **StunAttrString**
- struct **StunAttrIntegrity**
- struct **StunMessage**

### Defines

- #define **STUN\_MAX\_STRING** 256
- #define **STUN\_MAX\_UNKNOWN\_ATTRIBUTES** 8
- #define **STUN\_MAX\_MESSAGE\_SIZE** 2048
- #define **STUN\_PORT** 3478
- #define **TYPE\_BINDREQUESTMSG** 0x0001
- #define **TYPE\_BINDRESPONSEMSG** 0x0101
- #define **TYPE\_BINDERRORRESPONSEMSG** 0x0111
- #define **TYPE\_SHAREDSECRETREQUESTMSG** 0x0002
- #define **TYPE\_SHAREDSECRETRESPONSEMSG** 0x0102
- #define **TYPE\_SHAREDSECRETERORRESPONSEMSG** 0x0112
- #define **ATTR\_MAPPEDADDRESS** 0x0001
- #define **ATTR\_RESPONSEADDRESS** 0x0002
- #define **ATTR\_CHANGEREQUEST** 0x0003
- #define **ATTR\_SOURCEADDRESS** 0x0004
- #define **ATTR\_CHANGEDADDRESS** 0x0005
- #define **ATTR\_USERNAME** 0x0006
- #define **ATTR\_PASSWORD** 0x0007
- #define **ATTR\_MESSAGEINTEGRITY** 0x0008
- #define **ATTR\_ERRORCODE** 0x0009
- #define **ATTR\_UNKNOWNATTRIBUTE** 0x000A
- #define **ATTR\_REFLECTEDFROM** 0x000B
- #define **ATTR\_XORMAPPEDADDRESS** 0x0020
- #define **ATTR\_XORONLY** 0x0021
- #define **ATTR\_SERVERNAME** 0x0022
- #define **ATTR\_SECONDARYADDRESS** 0x0050
- #define **FAMILY\_IPV4** 0x01
- #define **FAMILY\_IPV6** 0x02
- #define **CHANGEPORT** 0x02
- #define **CHANGEIP** 0x04

## Functions

- int **StunParseMessage** (char \*buf, unsigned int bufLen, **StunMessage** \*message)
  - void **StunBuildSimpleRequest** (**StunMessage** \*msg, **StunAttrString** \*username, int change\_port, int change\_address, unsigned int id)
  - unsigned int **StunEncodeMessage** (**StunMessage** \*message, char \*buf, unsigned int bufLen, **StunAttrString** \*password)
  - void **StunCreateUserName** (**StunAddress4** \*addr, **StunAttrString** \*username)
  - void **StunCreatePassword** (**StunAttrString** \*username, **StunAttrString** \*password)
  - int **StunServerProcessMsg** (char \*buf, unsigned int bufLen, **StunAddress4** \*from, **StunAddress4** \*secondary, **StunAddress4** \*pri, **StunAddress4** \*alt, **StunMessage** \*resp, **StunAddress4** \*destination, **StunAttrString** \*password, int \*change\_port, int \*change\_address)
-

## Define Documentation

```
#define STUN_MAX_STRING 256

#define STUN_MAX_UNKNOWN_ATTRIBUTES 8

#define STUN_MAX_MESSAGE_SIZE 2048

#define STUN_PORT 3478

#define TYPE_BINDREQUESTMSG 0x0001

#define TYPE_BINDRESPONSEMSG 0x0101

#define TYPE_BINDERRORRESPONSEMSG 0x0111

#define TYPE_SHAREDSECRETREQUESTMSG 0x0002

#define TYPE_SHAREDSECRETRESPONSEMSG 0x0102

#define TYPE_SHAREDSECRETERESPONSEMSG 0x0112

#define ATTR_MAPPEDADDRESS 0x0001

#define ATTR_RESPONSEADDRESS 0x0002

#define ATTR_CHANGEREQUEST 0x0003

#define ATTR_SOURCEADDRESS 0x0004

#define ATTR_CHANGEDADDRESS 0x0005

#define ATTR_USERNAME 0x0006

#define ATTR_PASSWORD 0x0007

#define ATTR_MESSAGEINTEGRITY 0x0008

#define ATTR_ERRORCODE 0x0009

#define ATTR_UNKNOWNATTRIBUTE 0x000A

#define ATTR_REFLECTEDFROM 0x000B

#define ATTR_XORMAPPEDADDRESS 0x0020

#define ATTR_XORONLY 0x0021
```

```
#define ATTR_SERVERNAME 0x0022

#define ATTR_SECONDARYADDRESS 0x0050

#define FAMILY_IPV4 0x01

#define FAMILY_IPV6 0x02

#define CHANGEPORT 0x02

#define CHANGEIP 0x04
```

---

## Function Documentation

**int StunParseMessage (char \* *buf*, unsigned int *bufLen*, StunMessage \* *message*)**

Parse the received packet into a STUN message.

**Parameters:**

*buf* Pointer to input buffer containing raw STUN packet.

*bufLen* Length of input buffer.

*message* Parsed STUN message.

**Returns:**

Returns 0 on success, or a non-zero value on error.

**void StunBuildSimpleRequest (StunMessage \* *msg*, StunAttrString \* *username*, int *change\_port*, int *change\_address*, unsigned int *id*)**

Build a simple STUN BINDING request to determine the NAT address and/or port.

**Parameters:**

*msg* STUN message.

*username* Add username attribute. If the length of the username attribute is zero, then the attribute is not added to the STUN packet.

*change\_port* If non-zero, a CHANGE-PORT request is set in the STUN packet.

*change\_address* If non-zero, a CHANGE-ADDRESS request is set in the STUN packet.

*id* Message ID. Should be unique in the session.

**unsigned int StunEncodeMessage (StunMessage \* *message*, char \* *buf*, unsigned int *bufLen*, StunAttrString \* *password*)**

Encode STUN message into raw STUN packet suitable for sending on the wire.

**Parameters:**

*message* STUN message.

*buf* Output buffer to place the encoded message.

*bufLen* Length of output buffer.

*password* If the password length is non-zero, then the STUN packet is protected with an INTEGRITY attribute containing the SHA-1 HMAC derived from the password.

**Returns:**

Returns the length of the encoded packet.

**void StunCreateUserName (StunAddress4 \* *addr*, StunAttrString \* *username*)**

Create a username for the specified address. This username is used to uniquely identify the STUN session and is commonly used within the ICE algorithm.

**Parameters:**

*addr* Address/port to uniquely identify the session on the local host.

*username* Unique username (ASCII string).

**void StunCreatePassword (StunAttrString \* *username*, StunAttrString \* *password*)**

Create a password for the specified address. This password is used to protect the STUN session from tampering and is commonly used within the ICE algorithm.

**Parameters:**

*username* Unique username (ASCII string).

*password* Unique password (8-character ASCII string).

**int StunServerProcessMsg (char \* *buf*, unsigned int *bufLen*, StunAddress4 \* *from*, StunAddress4 \* *secondary*, StunAddress4 \* *pri*, StunAddress4 \* *alt*, StunMessage \* *resp*, StunAddress4 \* *destination*, StunAttrString \* *password*, int \* *change\_port*, int \* *change\_address*)**

Processes STUN requests as a STUN server and constructs the appropriate response.

**Parameters:**

*buf* Input buffer containing raw STUN packet.

*bufLen* Length of input buffer.

*from* Address of interface which the packet arrived on.

*secondary* Address of the other interface to include in the SECONDARY-ADDRESS attribute, if the length is non-zero.

*pri* Local primary address.

*alt* Local alternate address.

*resp* Generated STUN response.

*destination* Destination address to send response.

*password* Password for INTEGRITY attribute to protect packet.

*change\_port* Pointer to boolean to indicate whether to change the local port when sending the response.

*change\_address* Pointer to boolean to indicate whether to change the local address when sending the response.

**Returns:**

Returns zero on success, or a non-zero value on error.

## stunio.h File Reference

### stunio.hDefines

- `#define NATTYPE_UNKNOWN 0`
- `#define NATTYPE_OPEN 1`
- `#define NATTYPE_CONENAT 2`
- `#define NATTYPE_RESTRICTEDNAT 3`
- `#define NATTYPE_PORTRESTRICTEDNAT 4`
- `#define NATTYPE_SYMMETRICNAT 5`
- `#define NATTYPE_SYMMETRICFIREWALL 6`
- `#define NATTYPE_BLOCKED 7`
- `#define NATTYPE_FAILURE 8`

### Functions

- `void StunTest (StunAddress4 *dest, int testNum, StunAddress4 *srcAddr)`
- `int StunNatType (StunAddress4 *dest, int port, int *preservePort, int *hairpin, StunAddress4 *srcAddr)`
- `int StunOpenSocket (StunAddress4 *dest, StunAddress4 *srcAddr, int port, StunAddress4 *mappedAddr)`
- `int StunOpenSocketPair (StunAddress4 *dest, StunAddress4 *srcAddr, int srcPort, int *fd1, int *fd2, StunAddress4 *mappedAddr)`

---

### Define Documentation

**`#define NATTYPE_UNKNOWN 0`**

NAT types returned by `StunNatType()`.

**`#define NATTYPE_OPEN 1`**

**`#define NATTYPE_CONENAT 2`**

**`#define NATTYPE_RESTRICTEDNAT 3`**

**`#define NATTYPE_PORTRESTRICTEDNAT 4`**

**`#define NATTYPE_SYMMETRICNAT 5`**

**`#define NATTYPE_SYMMETRICFIREWALL 6`**

**`#define NATTYPE_BLOCKED 7`**

**`#define NATTYPE_FAILURE 8`**

---

## Function Documentation

**void StunTest (StunAddress4 \* *dest*, int *testNum*, StunAddress4 \* *srcAddr*)**

Quickly determine the external NAT address and port.

**Parameters:**

*dest* Destination or STUN server address.

*testNum* STUN test to perform.

*srcAddr* Local address to use when connecting to destination. On return, contains the external NAT address and port.

**int StunNatType (StunAddress4 \* *dest*, int *port*, int \* *preservePort*, int \* *hairpin*, StunAddress4 \* *srcAddr*)**

Determine as much information as possible about the NAT.

**Parameters:**

*dest* Destination or STUN server address.

*port* Local port to use when connecting to destination.

*preservePort* Pointer to boolean which specifies whether the NAT preserves the local port (ie NAT, not PAT)

*hairpin* Pointer to boolean which specifies whether the NAT will route internal packets addressed to its external interface address.

*srcAddr* Local address to use when connecting to destination. On return, contains the external NAT address and port.

**int StunOpenSocket (StunAddress4 \* *dest*, StunAddress4 \* *srcAddr*, int *port*, StunAddress4 \* *mappedAddr*)**

Open a socket to a destination address and return the external NAT address

**Parameters:**

*dest* Destination or STUN server address.

*srcAddr* Local address to use when connecting to destination.

*port* Local port to use when connecting to destination.

*mappedAddr* External NAT address and port.

**int StunOpenSocketPair (StunAddress4 \* *dest*, StunAddress4 \* *srcAddr*, int *srcPort*, int \* *fd1*, int \* *fd2*, StunAddress4 \* *mappedAddr*)**

Try to open a pair of sockets having consecutive external ports.

**Parameters:**

*dest* Destination or STUN server address.

*srcAddr* Local address to use when connecting to destination.

*srcPort* Local port to use when connecting to destination.

*fd1* First socket.

*fd2* Second socket.

*mappedAddr* External NAT address and lowest port for pair.

**Returns:**

Returns zero on success, or a non-zero value on error.

