

# libsep Reference Manual

1.0.0

Generated by Doxygen 1.5.2

Thu Apr 26 11:15:42 2007



# Contents

<b>1</b>	<b>libsep Data Structure Index</b>	<b>1</b>
1.1	libsep Data Structures . . . . .	1
<b>2</b>	<b>libsep File Index</b>	<b>3</b>
2.1	libsep File List . . . . .	3
<b>3</b>	<b>libsep Data Structure Documentation</b>	<b>5</b>
3.1	sep_header_t Struct Reference . . . . .	5
3.2	sep_fics_t Struct Reference . . . . .	7
3.3	sep_pkt_t Struct Reference . . . . .	8
<b>4</b>	<b>libsep File Documentation</b>	<b>9</b>
4.1	include/sep.h File Reference . . . . .	9
4.2	src/sep.c File Reference . . . . .	22



# Chapter 1

## libsep Data Structure Index

### 1.1 libsep Data Structures

Here are the data structures with brief descriptions:

<b>sep_header_t</b> (The SEP header) . . . . .	5
<b>sep_fics_t</b> (SEP FICS information) . . . . .	7
<b>sep_pkt_t</b> (A SEP packet) . . . . .	8



# Chapter 2

## libsep File Index

### 2.1 libsep File List

Here is a list of all files with brief descriptions:

include/sep.h . . . . .	9
src/sep.c . . . . .	22



# Chapter 3

## libsep Data Structure Documentation

### 3.1 sep\_header\_t Struct Reference

The SEP header.

#### Data Fields

- `uint16_t version`
- `char protocol_name [SEP_HDR_PROTO_NAME_SIZE]`
- `uint16_t se_id`
- `uint16_t flags`
- `uint16_t type`
- `uint16_t result_cc`
- `uint16_t result_rc`
- `char dst_addr [SEP_HDR_DST_ADDR_SIZE]`
- `char src_addr [SEP_HDR_SRC_ADDR_SIZE]`
- `char rrn [SEP_HDR_RRN_SIZE]`

#### 3.1.1 Detailed Description

The SEP header.

### 3.1.2 Field Documentation

3.1.2.1 `uint16_t sep_header_t::version`

3.1.2.2 `char sep_header_t::protocol_name[SEP_HDR_PROTO_NAME_SIZE]`

3.1.2.3 `uint16_t sep_header_t::se_id`

3.1.2.4 `uint16_t sep_header_t::flags`

3.1.2.5 `uint16_t sep_header_t::type`

3.1.2.6 `uint16_t sep_header_t::result_cc`

3.1.2.7 `uint16_t sep_header_t::result_rc`

3.1.2.8 `char sep_header_t::dst_addr[SEP_HDR_DST_ADDR_SIZE]`

3.1.2.9 `char sep_header_t::src_addr[SEP_HDR_SRC_ADDR_SIZE]`

3.1.2.10 `char sep_header_t::rrn[SEP_HDR_RRN_SIZE]`

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

- `include/sep.h`

## 3.2 sep\_fics\_t Struct Reference

SEP FICS information.

### Data Fields

- char **name** [SEP\_FICS\_LEN]  
*Name of the FICS.*
- int **size**  
*Size of the asociated data.*

### 3.2.1 Detailed Description

SEP FICS information.

### 3.2.2 Field Documentation

#### 3.2.2.1 char sep\_fics\_t::name[SEP\_FICS\_LEN]

Name of the FICS.

#### 3.2.2.2 int sep\_fics\_t::size

Size of the asociated data.

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

- include/sep.h

### 3.3 sep\_pkt\_t Struct Reference

A SEP packet.

#### Data Fields

- sep\_header\_t \* **hdr**
- unsigned long **len**
- char \* **data**

#### 3.3.1 Detailed Description

A SEP packet.

#### 3.3.2 Field Documentation

**3.3.2.1 sep\_header\_t\* sep\_pkt\_t::hdr**

**3.3.2.2 unsigned long sep\_pkt\_t::len**

**3.3.2.3 char\* sep\_pkt\_t::data**

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

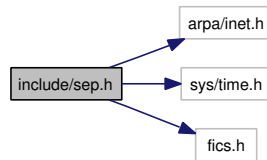
- include/**sep.h**

# Chapter 4

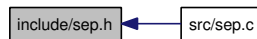
## libsep File Documentation

### 4.1 include/sep.h File Reference

Include dependency graph for sep.h:



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



### Data Structures

- struct **sep\_header\_t**  
*The SEP header.*
- struct **sep\_fics\_t**  
*SEP FICS information.*
- struct **sep\_pkt\_t**  
*A SEP packet.*

### Defines

- #define **SEP\_STX** 0x02
- #define **SEP\_ETX** 0x03
- #define **SEP\_DATA\_SEPARATOR** 0x09

- #define **SEP\_KEY\_VAL\_SEPARATOR** '='
- #define **SEP\_PROTOCOL\_NAME** "SEP "
- #define **SEP\_PROTOCOL\_VERSION** 1
- #define **SEP\_REG\_NUM\_SERVICE\_KEY** "NSVC"
- #define **SEP\_REG\_SERVICE\_KEY\_PFX** "SVC"
- #define **SEP\_FLAG\_REG** 0x01
- #define **SEP\_FLAG\_KEEPLIVE** 0x02
- #define **SEP\_FLAG\_REPLY** 0x10
- #define **SEP\_TYPE\_NUL** 0x00
- #define **SEP\_TYPE\_CRD** 0x01
- #define **SEP\_TYPE\_CUP** 0x02
- #define **SEP\_HDR\_PROTO\_NAME\_SIZE** 4
- #define **SEP\_HDR\_DST\_ADDR\_SIZE** 8
- #define **SEP\_HDR\_SRC\_ADDR\_SIZE** 8
- #define **SEP\_HDR\_RRN\_SIZE** 12
- #define **SEP\_HEADER\_SIZE** sizeof(sep\_header\_t)
- #define **SEP\_DATA\_OFFSET** (sizeof(sep\_header\_t) + 1)
- #define **SEP\_EMPTY\_PKT\_SIZE** (1 + SEP\_HEADER\_SIZE + 1)
- #define **SEP\_FICS\_LEN** FICS\_NAME\_LEN

## Enumerations

- enum {  
**SEP\_DISCONNECT = 1, SEP\_COMPLETE\_PKT, SEP\_NO\_PKT, SEP\_SENT\_PKT,**  
**SEP\_NO\_PKT\_SENT }**

*Return values from `sep_send_pkt()` (p. 19) and `sep_recv_pkt()` (p. 19).*

## Functions

- **int sep\_recv\_pkt** (int fd, **sep\_pkt\_t** \*pkt, int maxsize)  
*Tries to receive a SEP packet from a socket (non-blocking).*
- **int sep\_send\_pkt** (int fd, **sep\_pkt\_t** \*pkt)  
*Tries to send a SEP packet to a socket (non-blocking).*
- **void sep\_fics\_reset** (**sep\_fics\_t** \*fics)  
*Resets a `sep_fics_t` (p. 7) structure.*
- **char \* sep\_fics\_get** (**sep\_pkt\_t** \*pkt, **sep\_fics\_t** \*fics, const char \*name)  
*Tries to find a named FICS in a SEP packet.*
- **char \* sep\_fics\_get\_next** (**sep\_pkt\_t** \*pkt, **sep\_fics\_t** \*fics)  
*Tries to get the next FICS from a SEP packet.*
- **sep\_pkt\_t \* sep\_fics\_add** (**sep\_pkt\_t** \*pkt, const char \*name, const char \*data, int datalen)  
*Adds a FICS to a SEP packet.*
- **sep\_pkt\_t \* sep\_alloc\_pkt** (int len)

*Allocates memory for a SEP packet.*

- **sep\_pkt\_t \* sep\_alloc\_empty\_pkt** (uint16\_t seid, uint16\_t flags, uint16\_t type, uint16\_t cc, uint16\_t rsn, const char \*src\_comp, const char \*src\_func, const char \*dst\_comp, const char \*dst\_func, const char \*rrn)

*Allocates memory for an empty SEP packet.*

- **sep\_pkt\_t \* sep\_clone\_pkt\_wo\_ficse** (sep\_pkt\_t \*oldpkt)

*Clones a SEP packet and removes all FICSEs.*

- **sep\_pkt\_t \* sep\_free\_pkt** (sep\_pkt\_t \*pkt)

*Free a SEP packet.*

- void **sep\_set\_seid** (sep\_pkt\_t \*pkt, uint16\_t seid)

*Sets the SE ID field in the SEP header.*

- void **sep\_set\_flags** (sep\_pkt\_t \*pkt, uint16\_t flags)

*Sets the flags field in the SEP header.*

- void **sep\_set\_type** (sep\_pkt\_t \*pkt, uint16\_t type)

*Sets the type field in the SEP header.*

- void **sep\_set\_cc\_and\_rsn** (sep\_pkt\_t \*pkt, uint16\_t cc, uint16\_t rsn)

*Sets the result\_cc and result\_rc fields in the SEP header.*

- void **sep\_set\_src\_comp** (sep\_pkt\_t \*pkt, const char \*src\_comp)

*Sets the source component in the SEP header.*

- void **sep\_set\_src\_func** (sep\_pkt\_t \*pkt, const char \*src\_func)

*Sets the source function in the SEP header.*

- void **sep\_set\_dst\_comp** (sep\_pkt\_t \*pkt, const char \*dst\_comp)

*Sets the destination component in the SEP header.*

- void **sep\_set\_dst\_func** (sep\_pkt\_t \*pkt, const char \*dst\_func)

*Sets the destination function in the SEP header.*

- void **sep\_set\_rrn** (sep\_pkt\_t \*pkt, const char \*rrn)

*Sets the RRN field in the SEP header.*

- uint16\_t **sep\_get\_seid** (sep\_pkt\_t \*pkt)

*Returns the given packets SE ID.*

- uint16\_t **sep\_get\_flags** (sep\_pkt\_t \*pkt)

*Returns the given packets flags.*

- uint16\_t **sep\_get\_type** (sep\_pkt\_t \*pkt)

*Returns the given packets type.*

- uint16\_t **sep\_get\_cc** (sep\_pkt\_t \*pkt)

*Returns the given packets completion code.*

- `uint16_t sep_get_rsn (sep_pkt_t *pkt)`

*Returns the given packets reason code.*

- `char * sep_get_src_comp (sep_pkt_t *pkt)`

*Returns the given packets source component.*

- `char * sep_get_src_func (sep_pkt_t *pkt)`

*Returns the given packets source function.*

- `char * sep_get_dst_comp (sep_pkt_t *pkt)`

*Returns the given packets destination component.*

- `char * sep_get_dst_func (sep_pkt_t *pkt)`

*Returns the given packets destination function.*

- `uint32_t sep_get_rrn (sep_pkt_t *pkt)`

*Returns the given packets RRN as a 32-bit unsigned integer.*

- `char * sep_get_rrn_str (sep_pkt_t *pkt)`

*Returns the given packets RRN.*

### 4.1.1 Define Documentation

4.1.1.1 `#define SEP_DATA_OFFSET (sizeof(sep_header_t) + 1)`

4.1.1.2 `#define SEP_DATA_SEPARATOR 0x09`

4.1.1.3 `#define SEP_EMPTY_PKT_SIZE (1 + SEP_HEADER_SIZE + 1)`

4.1.1.4 `#define SEP_ETX 0x03`

4.1.1.5 `#define SEP_FICS_LEN FICS_NAME_LEN`

4.1.1.6 `#define SEP_FLAG_KEEPALIVE 0x02`

4.1.1.7 `#define SEP_FLAG_REG 0x01`

4.1.1.8 `#define SEP_FLAG_REPLY 0x10`

4.1.1.9 `#define SEP_HDR_DST_ADDR_SIZE 8`

4.1.1.10 `#define SEP_HDR_PROTO_NAME_SIZE 4`

4.1.1.11 `#define SEP_HDR_RRN_SIZE 12`

4.1.1.12 `#define SEP_HDR_SRC_ADDR_SIZE 8`

4.1.1.13 `#define SEP_HEADER_SIZE sizeof(sep_header_t)`

4.1.1.14 `#define SEP_KEY_VAL_SEPARATOR '='`

4.1.1.15 `#define SEP_PROTOCOL_NAME "SEP "`

4.1.1.16 `#define SEP_PROTOCOL_VERSION 1`

4.1.1.17 `#define SEP_REG_NUM_SERVICE_KEY "NSVC"`

4.1.1.18 `#define SEP_REG_SERVICE_KEY_PFX "SVC"`

4.1.1.19 `#define SEP_STX 0x02`

4.1.1.20 `#define SEP_TYPE_CRD 0x01`

4.1.1.21 `#define SEP_TYPE_CUP 0x02`

4.1.1.22 `#define SEP_TYPE_NUL 0x00`

### 4.1.2 Enumeration Type Documentation

#### 4.1.2.1 anonymous enum

Return values from `sep_send_pkt()` (p. 19) and `sep_rcv_pkt()` (p. 19).

**Enumerator:**

*SEP\_DISCONNECT*  
*SEP\_COMPLETE\_PKT*  
*SEP\_NO\_PKT*  
*SEP\_SENT\_PKT*  
*SEP\_NO\_PKT\_SENT*

**4.1.3 Function Documentation**

**4.1.3.1** `sep_pkt_t* sep_alloc_empty_pkt (uint16_t seid, uint16_t flags, uint16_t type, uint16_t cc, uint16_t rsn, const char * src_comp, const char * src_func, const char * dst_comp, const char * dst_func, const char * rrn)`

Allocates memory for an empty SEP packet.

**Parameters:**

*seid* the SE ID  
*flags* SEP flags; valid flags are defined with names SEP\_FLAG\_\*  
*type* SEP type; valid types are defined with names SEP\_TYPE\_\*  
*cc* the completion code  
*rsn* the reason code  
*src\_comp* source component  
*src\_func* source function  
*dst\_comp* destination component  
*dst\_func* destination function  
*rrn* the RRN

**Returns:**

A pointer to the allocated SEP packet.

**4.1.3.2** `sep_pkt_t* sep_alloc_pkt (int len)`

Allocates memory for a SEP packet.

The new packet has a STX, an empty header and an ETX.

**Parameters:**

*len* length of the packet (including STX, header and ETX)

**Returns:**

A pointer to the allocated SEP packet.

#### 4.1.3.3 `sep_pkt_t* sep_clone_pkt_wo_ficses (sep_pkt_t * oldpkt)`

Clones a SEP packet and removes all FICSes.

**Parameters:**

*oldpkt* the packet to clone

**Returns:**

The new SEP packet.

#### 4.1.3.4 `sep_pkt_t* sep_fics_add (sep_pkt_t * pkt, const char * name, const char * data, int datalen)`

Adds a FICS to a SEP packet.

**Parameters:**

*pkt* the SEP packet

*name* name of the FICS

*data* the FICS data

*datalen* length of the FICS data

**Returns:**

A pointer to the new packet. The old packet is free():d.

#### 4.1.3.5 `char* sep_fics_get (sep_pkt_t * pkt, sep_fics_t * fics, const char * name)`

Tries to find a named FICS in a SEP packet.

Call `sep_fics_reset()` (p. 16) on the `sep_fics_t` (p. 7) before passing it to this function.

**Parameters:**

*pkt* the SEP packet

*fics* pointer to a `sep_fics_t` (p. 7) which will be filled

*name* name of the FICS

**Returns:**

Pointer to the FICS data if FICS name was found; NULL otherwise.

#### 4.1.3.6 `char* sep_fics_get_next (sep_pkt_t * pkt, sep_fics_t * fics)`

Tries to get the next FICS from a SEP packet.

Call `sep_fics_reset()` (p. 16) before the first call to this function. The function is `_NOT_` reentrant.

**Parameters:**

*pkt* the SEP packet

*fics* pointer to a `sep_fics_t` (p. 7) which will be filled

**Returns:**

Pointer to the FICS data if FICS name was found; NULL otherwise.

**4.1.3.7 void sep\_fics\_reset (sep\_fics\_t \* *fics*)**

Resets a `sep_fics_t` (p. 7) structure.

**Parameters:**

*fics* pointer to the structure

**4.1.3.8 sep\_pkt\_t\* sep\_free\_pkt (sep\_pkt\_t \* *pkt*)**

Free a SEP packet.

**Parameters:**

*pkt* pointer to the packet which should be free:d

**Returns:**

always NULL

**4.1.3.9 uint16\_t sep\_get\_cc (sep\_pkt\_t \* *pkt*)**

Returns the given packets completion code.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The completion code.

**4.1.3.10 char\* sep\_get\_dst\_comp (sep\_pkt\_t \* *pkt*)**

Returns the given packets destination component.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The destination component.

**4.1.3.11 char\* sep\_get\_dst\_func (sep\_pkt\_t \* pkt)**

Returns the given packets destination function.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The destination component.

**4.1.3.12 uint16\_t sep\_get\_flags (sep\_pkt\_t \* pkt)**

Returns the given packets flags.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The flags.

**4.1.3.13 uint32\_t sep\_get\_rrn (sep\_pkt\_t \* pkt)**

Returns the given packets RRN as a 32-bit unsigned integer.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The RRN.

**4.1.3.14 char\* sep\_get\_rrn\_str (sep\_pkt\_t \* pkt)**

Returns the given packets RRN.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The RRN.

**4.1.3.15** `uint16_t sep_get_rsn (sep_pkt_t * pkt)`

Returns the given packets reason code.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The reason code.

**4.1.3.16** `uint16_t sep_get_seid (sep_pkt_t * pkt)`

Returns the given packets SE ID.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The SE ID.

**4.1.3.17** `char* sep_get_src_comp (sep_pkt_t * pkt)`

Returns the given packets source component.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The source component.

**4.1.3.18** `char* sep_get_src_func (sep_pkt_t * pkt)`

Returns the given packets source function.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The source function.

**4.1.3.19 uint16\_t sep\_get\_type (sep\_pkt\_t \* pkt)**

Returns the given packets type.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The type.

**4.1.3.20 int sep\_recv\_pkt (int fd, sep\_pkt\_t \* pkt, int maxsize)**

Tries to receive a SEP packet from a socket (non-blocking).

**Parameters:**

*fd* the file descriptor

*pkt* the packet which received data should be stored in

*maxsize* receive at most maxsize bytes

**Returns:**

SEP\_DISCONNECT if recv returns <= 0, SEP\_COMPLETE\_PKT if a complete packet was received or SEP\_NO\_PKT if no packet was received.

**4.1.3.21 int sep\_send\_pkt (int fd, sep\_pkt\_t \* pkt)**

Tries to send a SEP packet to a socket (non-blocking).

**Parameters:**

*fd* the file descriptor

*pkt* the packet which received data should be stored in

**Returns:**

SEP\_DISCONNECT if send returns <= 0, SEP\_SENT\_PKT if a packet was sent or SEP\_NO\_PKT\_SENT if no packet was sent.

**4.1.3.22 void sep\_set\_cc\_and\_rsn (sep\_pkt\_t \* pkt, uint16\_t cc, uint16\_t rsn)**

Sets the result\_cc and result\_rc fields in the SEP header.

**Parameters:**

*pkt* the SEP packet

*cc* the new completion code

*rsn* the new reason code

**4.1.3.23 void sep\_set\_dst\_comp (sep\_pkt\_t \* *pkt*, const char \* *dst\_comp*)**

Sets the destination component in the SEP header.

**Parameters:**

*pkt* the SEP packet

*dst\_comp* the new destination component

**4.1.3.24 void sep\_set\_dst\_func (sep\_pkt\_t \* *pkt*, const char \* *dst\_func*)**

Sets the destination function in the SEP header.

**Parameters:**

*pkt* the SEP packet

*dst\_func* the new destination function

**4.1.3.25 void sep\_set\_flags (sep\_pkt\_t \* *pkt*, uint16\_t *flags*)**

Sets the flags field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*flags* the new flags

**4.1.3.26 void sep\_set\_rrn (sep\_pkt\_t \* *pkt*, const char \* *rrn*)**

Sets the RRN field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*rrn* the new RRN

**4.1.3.27 void sep\_set\_seid (sep\_pkt\_t \* *pkt*, uint16\_t *seid*)**

Sets the SE ID field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*seid* the new SE ID

**4.1.3.28 void sep\_set\_src\_comp (sep\_pkt\_t \* *pkt*, const char \* *src\_comp*)**

Sets the source component in the SEP header.

**Parameters:**

*pkt* the SEP packet

*src\_comp* the new source component

**4.1.3.29 void sep\_set\_src\_func (sep\_pkt\_t \* *pkt*, const char \* *src\_func*)**

Sets the source function in the SEP header.

**Parameters:**

*pkt* the SEP packet

*src\_func* the new source function

**4.1.3.30 void sep\_set\_type (sep\_pkt\_t \* *pkt*, uint16\_t *type*)**

Sets the type field in the SEP header.

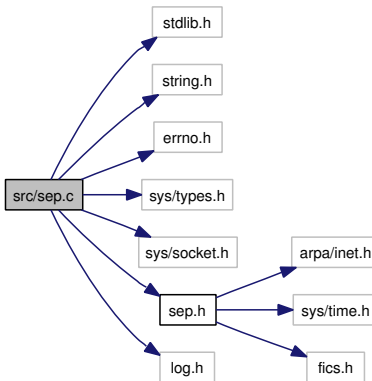
**Parameters:**

*pkt* the SEP packet

*type* the new type

## 4.2 src/sep.c File Reference

Include dependency graph for sep.c:



### Functions

- **int sep\_rcv\_pkt** (int fd, **sep\_pkt\_t** \*pkt, int maxsize)  
*Tries to receive a SEP packet from a socket (non-blocking).*
- **int sep\_send\_pkt** (int fd, **sep\_pkt\_t** \*pkt)  
*Tries to send a SEP packet to a socket (non-blocking).*
- **void sep\_fics\_reset** (**sep\_fics\_t** \*fics)  
*Resets a **sep\_fics\_t** (p. 7) structure.*
- **char \* sep\_fics\_get** (**sep\_pkt\_t** \*pkt, **sep\_fics\_t** \*fics, const char \*name)  
*Tries to find a named FICS in a SEP packet.*
- **char \* sep\_fics\_get\_next** (**sep\_pkt\_t** \*pkt, **sep\_fics\_t** \*fics)  
*Tries to get the next FICS from a SEP packet.*
- **sep\_pkt\_t \* sep\_fics\_add** (**sep\_pkt\_t** \*pkt, const char \*name, const char \*data, int datalen)  
*Adds a FICS to a SEP packet.*
- **sep\_pkt\_t \* sep\_alloc\_pkt** (int len)  
*Allocates memory for a SEP packet.*
- **sep\_pkt\_t \* sep\_alloc\_empty\_pkt** (uint16\_t seid, uint16\_t flags, uint16\_t type, uint16\_t cc, uint16\_t rsn, const char \*src\_comp, const char \*src\_func, const char \*dst\_comp, const char \*dst\_func, const char \*rrn)  
*Allocates memory for an empty SEP packet.*
- **sep\_pkt\_t \* sep\_clone\_pkt\_wo\_fices** (**sep\_pkt\_t** \*oldpkt)  
*Clones a SEP packet and removes all FICSES.*
- **sep\_pkt\_t \* sep\_free\_pkt** (**sep\_pkt\_t** \*pkt)

*Free a SEP packet.*

- void **sep\_set\_seid** (**sep\_pkt\_t** \*pkt, uint16\_t seid)  
*Sets the SE ID field in the SEP header.*
- void **sep\_set\_flags** (**sep\_pkt\_t** \*pkt, uint16\_t flags)  
*Sets the flags field in the SEP header.*
- void **sep\_set\_type** (**sep\_pkt\_t** \*pkt, uint16\_t type)  
*Sets the type field in the SEP header.*
- void **sep\_set\_cc\_and\_rsn** (**sep\_pkt\_t** \*pkt, uint16\_t cc, uint16\_t rsn)  
*Sets the result\_cc and result\_rc fields in the SEP header.*
- void **sep\_set\_src\_comp** (**sep\_pkt\_t** \*pkt, const char \*src\_comp)  
*Sets the source component in the SEP header.*
- void **sep\_set\_src\_func** (**sep\_pkt\_t** \*pkt, const char \*src\_func)  
*Sets the source function in the SEP header.*
- void **sep\_set\_dst\_comp** (**sep\_pkt\_t** \*pkt, const char \*dst\_comp)  
*Sets the destination component in the SEP header.*
- void **sep\_set\_dst\_func** (**sep\_pkt\_t** \*pkt, const char \*dst\_func)  
*Sets the destination function in the SEP header.*
- void **sep\_set\_rrn** (**sep\_pkt\_t** \*pkt, const char \*rrn)  
*Sets the RRN field in the SEP header.*
- uint16\_t **sep\_get\_seid** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets SE ID.*
- uint16\_t **sep\_get\_flags** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets flags.*
- uint16\_t **sep\_get\_type** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets type.*
- uint16\_t **sep\_get\_cc** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets completion code.*
- uint16\_t **sep\_get\_rsn** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets reason code.*
- char \* **sep\_get\_src\_comp** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets source component.*
- char \* **sep\_get\_src\_func** (**sep\_pkt\_t** \*pkt)  
*Returns the given packets source function.*

- `char * sep_get_dst_comp (sep_pkt_t *pkt)`  
Returns the given packets destination component.
- `char * sep_get_dst_func (sep_pkt_t *pkt)`  
Returns the given packets destination function.
- `uint32_t sep_get_rrn (sep_pkt_t *pkt)`  
Returns the given packets RRN as a 32-bit unsigned integer.
- `char * sep_get_rrn_str (sep_pkt_t *pkt)`  
Returns the given packets RRN.

## 4.2.1 Function Documentation

### 4.2.1.1 `sep_pkt_t* sep_alloc_empty_pkt (uint16_t seid, uint16_t flags, uint16_t type, uint16_t cc, uint16_t rsn, const char * src_comp, const char * src_func, const char * dst_comp, const char * dst_func, const char * rrn)`

Allocates memory for an empty SEP packet.

#### Parameters:

*seid* the SE ID  
*flags* SEP flags; valid flags are defined with names SEP\_FLAG\_\*  
*type* SEP type; valid types are defined with names SEP\_TYPE\_\*  
*cc* the completion code  
*rsn* the reason code  
*src\_comp* source component  
*src\_func* source function  
*dst\_comp* destination component  
*dst\_func* destination function  
*rrn* the RRN

#### Returns:

A pointer to the allocated SEP packet.

### 4.2.1.2 `sep_pkt_t* sep_alloc_pkt (int len)`

Allocates memory for a SEP packet.

The new packet has a STX, an empty header and an ETX.

#### Parameters:

*len* length of the packet (including STX, header and ETX)

#### Returns:

A pointer to the allocated SEP packet.

#### 4.2.1.3 `sep_pkt_t* sep_clone_pkt_wo_ficses (sep_pkt_t * oldpkt)`

Clones a SEP packet and removes all FICSes.

**Parameters:**

*oldpkt* the packet to clone

**Returns:**

The new SEP packet.

#### 4.2.1.4 `sep_pkt_t* sep_fics_add (sep_pkt_t * pkt, const char * name, const char * data, int datalen)`

Adds a FICS to a SEP packet.

**Parameters:**

*pkt* the SEP packet

*name* name of the FICS

*data* the FICS data

*datalen* length of the FICS data

**Returns:**

A pointer to the new packet. The old packet is free():d.

#### 4.2.1.5 `char* sep_fics_get (sep_pkt_t * pkt, sep_fics_t * fics, const char * name)`

Tries to find a named FICS in a SEP packet.

Call `sep_fics_reset()` (p. 16) on the `sep_fics_t` (p. 7) before passing it to this function.

**Parameters:**

*pkt* the SEP packet

*fics* pointer to a `sep_fics_t` (p. 7) which will be filled

*name* name of the FICS

**Returns:**

Pointer to the FICS data if FICS name was found; NULL otherwise.

#### 4.2.1.6 `char* sep_fics_get_next (sep_pkt_t * pkt, sep_fics_t * fics)`

Tries to get the next FICS from a SEP packet.

Call `sep_fics_reset()` (p. 16) before the first call to this function. The function is `_NOT_` reentrant.

**Parameters:**

*pkt* the SEP packet

*fics* pointer to a `sep_fics_t` (p. 7) which will be filled

**Returns:**

Pointer to the FICS data if FICS name was found; NULL otherwise.

**4.2.1.7 void sep\_fics\_reset (sep\_fics\_t \* *fics*)**

Resets a `sep_fics_t` (p. 7) structure.

**Parameters:**

*fics* pointer to the structure

**4.2.1.8 sep\_pkt\_t\* sep\_free\_pkt (sep\_pkt\_t \* *pkt*)**

Free a SEP packet.

**Parameters:**

*pkt* pointer to the packet which should be free:d

**Returns:**

always NULL

**4.2.1.9 uint16\_t sep\_get\_cc (sep\_pkt\_t \* *pkt*)**

Returns the given packets completion code.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The completion code.

**4.2.1.10 char\* sep\_get\_dst\_comp (sep\_pkt\_t \* *pkt*)**

Returns the given packets destination component.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The destination component.

**4.2.1.11 char\* sep\_get\_dst\_func (sep\_pkt\_t \* pkt)**

Returns the given packets destination function.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The destination component.

**4.2.1.12 uint16\_t sep\_get\_flags (sep\_pkt\_t \* pkt)**

Returns the given packets flags.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The flags.

**4.2.1.13 uint32\_t sep\_get\_rrn (sep\_pkt\_t \* pkt)**

Returns the given packets RRN as a 32-bit unsigned integer.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The RRN.

**4.2.1.14 char\* sep\_get\_rrn\_str (sep\_pkt\_t \* pkt)**

Returns the given packets RRN.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The RRN.

**4.2.1.15** `uint16_t sep_get_rsn (sep_pkt_t * pkt)`

Returns the given packets reason code.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The reason code.

**4.2.1.16** `uint16_t sep_get_seid (sep_pkt_t * pkt)`

Returns the given packets SE ID.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The SE ID.

**4.2.1.17** `char* sep_get_src_comp (sep_pkt_t * pkt)`

Returns the given packets source component.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The source component.

**4.2.1.18** `char* sep_get_src_func (sep_pkt_t * pkt)`

Returns the given packets source function.

This string is `_NOT_NULL` terminated.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The source function.

**4.2.1.19 uint16\_t sep\_get\_type (sep\_pkt\_t \* pkt)**

Returns the given packets type.

**Parameters:**

*pkt* the SEP packet

**Returns:**

The type.

**4.2.1.20 int sep\_recv\_pkt (int fd, sep\_pkt\_t \* pkt, int maxsize)**

Tries to receive a SEP packet from a socket (non-blocking).

**Parameters:**

*fd* the file descriptor

*pkt* the packet which received data should be stored in

*maxsize* receive at most maxsize bytes

**Returns:**

SEP\_DISCONNECT if recv returns <= 0, SEP\_COMPLETE\_PKT if a complete packet was received or SEP\_NO\_PKT if no packet was received.

**4.2.1.21 int sep\_send\_pkt (int fd, sep\_pkt\_t \* pkt)**

Tries to send a SEP packet to a socket (non-blocking).

**Parameters:**

*fd* the file descriptor

*pkt* the packet which received data should be stored in

**Returns:**

SEP\_DISCONNECT if send returns <= 0, SEP\_SENT\_PKT if a packet was sent or SEP\_NO\_PKT\_SENT if no packet was sent.

**4.2.1.22 void sep\_set\_cc\_and\_rsn (sep\_pkt\_t \* pkt, uint16\_t cc, uint16\_t rsn)**

Sets the result\_cc and result\_rc fields in the SEP header.

**Parameters:**

*pkt* the SEP packet

*cc* the new completion code

*rsn* the new reason code

**4.2.1.23 void sep\_set\_dst\_comp (sep\_pkt\_t \* *pkt*, const char \* *dst\_comp*)**

Sets the destination component in the SEP header.

**Parameters:**

*pkt* the SEP packet

*dst\_comp* the new destination component

**4.2.1.24 void sep\_set\_dst\_func (sep\_pkt\_t \* *pkt*, const char \* *dst\_func*)**

Sets the destination function in the SEP header.

**Parameters:**

*pkt* the SEP packet

*dst\_func* the new destination function

**4.2.1.25 void sep\_set\_flags (sep\_pkt\_t \* *pkt*, uint16\_t *flags*)**

Sets the flags field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*flags* the new flags

**4.2.1.26 void sep\_set\_rrn (sep\_pkt\_t \* *pkt*, const char \* *rrn*)**

Sets the RRN field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*rrn* the new RRN

**4.2.1.27 void sep\_set\_seid (sep\_pkt\_t \* *pkt*, uint16\_t *seid*)**

Sets the SE ID field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*seid* the new SE ID

**4.2.1.28 void sep\_set\_src\_comp (sep\_pkt\_t \* *pkt*, const char \* *src\_comp*)**

Sets the source component in the SEP header.

**Parameters:**

*pkt* the SEP packet

*src\_comp* the new source component

**4.2.1.29 void sep\_set\_src\_func (sep\_pkt\_t \* *pkt*, const char \* *src\_func*)**

Sets the source function in the SEP header.

**Parameters:**

*pkt* the SEP packet

*src\_func* the new source function

**4.2.1.30 void sep\_set\_type (sep\_pkt\_t \* *pkt*, uint16\_t *type*)**

Sets the type field in the SEP header.

**Parameters:**

*pkt* the SEP packet

*type* the new type

# Index

- sep\_header\_t, 5
  - dst\_addr, 6
  - flags, 6
  - protocol\_name, 6
  - result\_cc, 6
  - result\_rc, 6
  - rrn, 6
  - se\_id, 6
  - src\_addr, 6
  - type, 6
  - version, 6
- data
  - sep\_pkt\_t, 8
- dst\_addr
  - sep\_header\_t, 6
- flags
  - sep\_header\_t, 6
- hdr
  - sep\_pkt\_t, 8
- include/sep.h, 9
- len
  - sep\_pkt\_t, 8
- name
  - sep\_fics\_t, 7
- protocol\_name
  - sep\_header\_t, 6
- result\_cc
  - sep\_header\_t, 6
- result\_rc
  - sep\_header\_t, 6
- rrn
  - sep\_header\_t, 6
- se\_id
  - sep\_header\_t, 6
- sep.c
  - sep\_alloc\_empty\_pkt, 24
  - sep\_alloc\_pkt, 24
  - sep\_clone\_pkt\_wo\_fices, 24
  - sep\_fics\_add, 25
  - sep\_fics\_get, 25
  - sep\_fics\_get\_next, 25
  - sep\_fics\_reset, 26
  - sep\_free\_pkt, 26
  - sep\_get\_cc, 26
  - sep\_get\_dst\_comp, 26
  - sep\_get\_dst\_func, 26
  - sep\_get\_flags, 27
  - sep\_get\_rrn, 27
  - sep\_get\_rrn\_str, 27
  - sep\_get\_rsn, 27
  - sep\_get\_seid, 28
  - sep\_get\_src\_comp, 28
  - sep\_get\_src\_func, 28
  - sep\_get\_type, 28
  - sep\_recv\_pkt, 29
  - sep\_send\_pkt, 29
  - sep\_set\_cc\_and\_rsn, 29
  - sep\_set\_dst\_comp, 29
  - sep\_set\_dst\_func, 30
  - sep\_set\_flags, 30
  - sep\_set\_rrn, 30
  - sep\_set\_seid, 30
  - sep\_set\_src\_comp, 30
  - sep\_set\_src\_func, 31
  - sep\_set\_type, 31
- sep.h
  - sep\_alloc\_empty\_pkt, 14
  - sep\_alloc\_pkt, 14
  - sep\_clone\_pkt\_wo\_fices, 14
  - SEP\_COMPLETE\_PKT, 14
  - SEP\_DATA\_OFFSET, 13
  - SEP\_DATA\_SEPARATOR, 13
  - SEP\_DISCONNECT, 13
  - SEP\_EMPTY\_PKT\_SIZE, 13
  - SEP\_ETX, 13
  - sep\_fics\_add, 15
  - sep\_fics\_get, 15
  - sep\_fics\_get\_next, 15
  - SEP\_FICS\_LEN, 13
  - sep\_fics\_reset, 16
  - SEP\_FLAG\_KEEPALIVE, 13
  - SEP\_FLAG\_REG, 13

SEP\_FLAG\_REPLY, 13  
sep\_free\_pkt, 16  
sep\_get\_cc, 16  
sep\_get\_dst\_comp, 16  
sep\_get\_dst\_func, 16  
sep\_get\_flags, 17  
sep\_get\_rrn, 17  
sep\_get\_rrn\_str, 17  
sep\_get\_rsn, 17  
sep\_get\_seid, 18  
sep\_get\_src\_comp, 18  
sep\_get\_src\_func, 18  
sep\_get\_type, 18  
SEP\_HDR\_DST\_ADDR\_SIZE, 13  
SEP\_HDR\_PROTO\_NAME\_SIZE, 13  
SEP\_HDR\_RRN\_SIZE, 13  
SEP\_HDR\_SRC\_ADDR\_SIZE, 13  
SEP\_HEADER\_SIZE, 13  
SEP\_KEY\_VAL\_SEPARATOR, 13  
SEP\_NO\_PKT, 14  
SEP\_NO\_PKT\_SENT, 14  
SEP\_PROTOCOL\_NAME, 13  
SEP\_PROTOCOL\_VERSION, 13  
sep\_rcv\_pkt, 19  
SEP\_REG\_NUM\_SERVICE\_KEY, 13  
SEP\_REG\_SERVICE\_KEY\_PFX, 13  
sep\_send\_pkt, 19  
SEP\_SENT\_PKT, 14  
sep\_set\_cc\_and\_rsn, 19  
sep\_set\_dst\_comp, 19  
sep\_set\_dst\_func, 20  
sep\_set\_flags, 20  
sep\_set\_rrn, 20  
sep\_set\_seid, 20  
sep\_set\_src\_comp, 20  
sep\_set\_src\_func, 21  
sep\_set\_type, 21  
SEP\_STX, 13  
SEP\_TYPE\_CRD, 13  
SEP\_TYPE\_CUP, 13  
SEP\_TYPE\_NUL, 13  
sep\_alloc\_empty\_pkt  
sep.c, 24  
sep.h, 14  
sep\_alloc\_pkt  
sep.c, 24  
sep.h, 14  
sep\_clone\_pkt\_wo\_ficses  
sep.c, 24  
sep.h, 14  
SEP\_COMPLETE\_PKT  
sep.h, 14  
SEP\_DATA\_OFFSET  
sep.h, 13  
SEP\_DATA\_SEPARATOR  
sep.h, 13  
SEP\_DISCONNECT  
sep.h, 13  
SEP\_EMPTY\_PKT\_SIZE  
sep.h, 13  
SEP\_ETX  
sep.h, 13  
sep\_fics\_add  
sep.c, 25  
sep.h, 15  
sep\_fics\_get  
sep.c, 25  
sep.h, 15  
sep\_fics\_get\_next  
sep.c, 25  
sep.h, 15  
SEP\_FICS\_LEN  
sep.h, 13  
sep\_fics\_reset  
sep.c, 26  
sep.h, 16  
sep\_fics\_t, 7  
name, 7  
size, 7  
SEP\_FLAG\_KEEPALIVE  
sep.h, 13  
SEP\_FLAG\_REG  
sep.h, 13  
SEP\_FLAG\_REPLY  
sep.h, 13  
sep\_free\_pkt  
sep.c, 26  
sep.h, 16  
sep\_get\_cc  
sep.c, 26  
sep.h, 16  
sep\_get\_dst\_comp  
sep.c, 26  
sep.h, 16  
sep\_get\_dst\_func  
sep.c, 26  
sep.h, 16  
sep\_get\_flags  
sep.c, 27  
sep.h, 17  
sep\_get\_rrn  
sep.c, 27  
sep.h, 17  
sep\_get\_rrn\_str  
sep.c, 27  
sep.h, 17  
sep\_get\_rsn  
sep.c, 27

---

sep.h, 17  
 sep\_get\_seid  
   sep.c, 28  
   sep.h, 18  
 sep\_get\_src\_comp  
   sep.c, 28  
   sep.h, 18  
 sep\_get\_src\_func  
   sep.c, 28  
   sep.h, 18  
 sep\_get\_type  
   sep.c, 28  
   sep.h, 18  
 SEP\_HDR\_DST\_ADDR\_SIZE  
   sep.h, 13  
 SEP\_HDR\_PROTO\_NAME\_SIZE  
   sep.h, 13  
 SEP\_HDR\_RRN\_SIZE  
   sep.h, 13  
 SEP\_HDR\_SRC\_ADDR\_SIZE  
   sep.h, 13  
 SEP\_HEADER\_SIZE  
   sep.h, 13  
 SEP\_KEY\_VAL\_SEPARATOR  
   sep.h, 13  
 SEP\_NO\_PKT  
   sep.h, 14  
 SEP\_NO\_PKT\_SENT  
   sep.h, 14  
 sep\_pkt\_t, 8  
   data, 8  
   hdr, 8  
   len, 8  
 SEP\_PROTOCOL\_NAME  
   sep.h, 13  
 SEP\_PROTOCOL\_VERSION  
   sep.h, 13  
 sep\_rcv\_pkt  
   sep.c, 29  
   sep.h, 19  
 SEP\_REG\_NUM\_SERVICE\_KEY  
   sep.h, 13  
 SEP\_REG\_SERVICE\_KEY\_PFX  
   sep.h, 13  
 sep\_send\_pkt  
   sep.c, 29  
   sep.h, 19  
 SEP\_SENT\_PKT  
   sep.h, 14  
 sep\_set\_cc\_and\_rsn  
   sep.c, 29  
   sep.h, 19  
 sep\_set\_dst\_comp  
   sep.c, 29  
   sep.h, 19  
 sep\_set\_dst\_func  
   sep.c, 30  
   sep.h, 20  
 sep\_set\_flags  
   sep.c, 30  
   sep.h, 20  
 sep\_set\_rrn  
   sep.c, 30  
   sep.h, 20  
 sep\_set\_seid  
   sep.c, 30  
   sep.h, 20  
 sep\_set\_src\_comp  
   sep.c, 30  
   sep.h, 20  
 sep\_set\_src\_func  
   sep.c, 31  
   sep.h, 21  
 sep\_set\_type  
   sep.c, 31  
   sep.h, 21  
 SEP\_STX  
   sep.h, 13  
 SEP\_TYPE\_CRD  
   sep.h, 13  
 SEP\_TYPE\_CUP  
   sep.h, 13  
 SEP\_TYPE\_NUL  
   sep.h, 13  
 size  
   sep\_fics\_t, 7  
 src/sep.c, 22  
 src\_addr  
   sep\_header\_t, 6  
  
 type  
   sep\_header\_t, 6  
  
 version  
   sep\_header\_t, 6