



HS SMTP Library v2.0.8

User Manual

Revision: 2.2

Date: 11 January 2012

HS SMTP LIBRARY V2.0.8.....	1
USER MANUAL.....	1
REVISION: 2.2.....	1
DATE: 11 JANUARY 2012.....	1
1 INTRODUCTION.....	2
2 HS SMTP API.....	3
2.1 HsSMTPINIT.....	3
2.2 HsSMTPDESTROY.....	3
2.3 HsSMTPTICK.....	3
2.4 HsSMTPSENDMAIL.....	4
2.5 HsSMTPABORTMAIL.....	5
2.6 HsSMTPCONTINUE.....	5
3 HS SMTP TO USER EVENT CALLBACK AND EVENTS.....	6
3.1 EVENT CALLBACK PROTOTYPE.....	6
3.2 EVENT CODES.....	6

1 Introduction

HS SMTP is a software library in C (supplied with full source code) which implements the client side of Simple Mail Transfer Protocol (SMTP) over TCP socket layer according to RFC 821. Support for transfer of basic message header and text is provided.

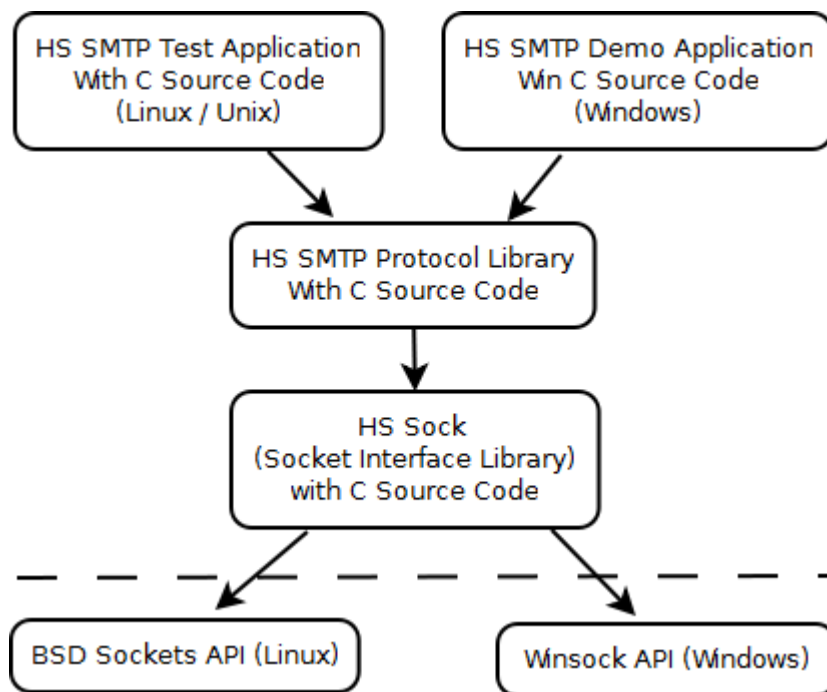
The same HS SMTP source code can be compiled for Windows and Linux target.

HS SMTP supports ESMTP extension for LOGIN Authentication using Base64 encoding and message sending to multiple recipients from address list.

HS SMTP supports sending binary file attachments using MIME version 1.0 base64 encoding.

HS SMTP supports SSL / TLS and secure SMTP over SSL / TLS (RFC 3207 using STARTTLS command) and SSL without STARTTLS

HS SMTP product includes SMTP protocol library, socket interface library and demo applications:



HS SMTP Library incorporates the necessary state machine, transparency procedures, and server response processing required to provide for a simple and robust SMTP client implementation

2 HS SMTP API

2.1 *HsSmtplnit*

Initialises HS SMTP Library.

Syntax:

```
int HsSmtplnit(void);
```

Parameters:

None

Returns:

- *HS SMTP_RC_OK* – success, HS SMTP Library successfully initialised
- *HS SMTP_RC_ALRINIT* – HS SMTP Library is already initialised
- *HS SMTP_RC_SOCKINIT_FAIL* – General socket interface module initialisation problem
- *HS SMTP_RC_SOCKINIT2_FAIL* – Socket layer could not get local hostname

Remarks:

This function must be called at initialisation, before any other functions are called. Calling HsSmtplnit twice will return an error. You can call HsSmtplDestroy first to de-initialise HS SMTP Library and then call HsSmtplnit again. While a library is initialized you can send mails any number of times calling HsSmtplSendMail

2.2 *HsSmtplDestroy*

De-Initialises HS SMTP Library.

Syntax:

```
int HsSmtplDestroy(void);
```

Parameters:

None

Returns:

- *HS SMTP_RC_OK* – success, HS SMTP Library successfully de-initialised
- *HS SMTP_RC_NOTINIT* – cannot destroy, HS SMTP Library not yet initialised

Remarks:

This function releases all used resources and cleans up Socket Interface Layer

2.3 *HsSmtplTick*

Drives the internal operations of HS SMTP library.

Syntax:

```
int HsSmtplTick(void);
```

Parameters:

None

Returns:

- *HS SMTP_RC_OK* – success
- *HS SMTP_RC_NOTINIT* – HS SMTP Library not initialised

Remarks:

This function must be called from the user application periodically and as often as possible. This function drives internal operation of the socket layer (reading events from TCP sockets) and pacing internal SMTP timers.

2.4 HsSmtplibSendMail

Initiates sending an email message to the remote SMTP server

Syntax:

```
int HsSmtplibSendMail(hs_smtplib_mail_t *m, long *session_handle);
```

Parameters:

hs_smtplib_mail_t *m – pointer to structure defined below:

Field name	Description
srv_name	Pointer to SMTP server name Max. length limited to 80 bytes (Note 1)
srv_ip	Pointer to SMTP IP address string in the format “n.n.n.n” (Note 1)
srv_port	Decimal SMTP server port number. (usually 25)
send_addr	Sender email address string Max. length limited to 80 bytes
unsigned char *recv_addr[50];	Array of maximum 50 Destination email address strings. Maximum length of an address is limited to 80 bytes
recv_addr_num	Number of valid elements in the recv_addr array – number of destination email addresses.
subj	Message subject. Max. length limited to 80 bytes
unsigned char *cc[50];	Array of maximum 50 cc email email address strings. Maximum length of an address is limited to 80 bytes
cc_addr_num	Number of valid elements in the cc array – number of destination CC email addresses.
msgp	Pointer to message text buffer
msglen	message text buffer length
do_auth	Integer. Set to 1 if SMTP server requires authentication. LOGIN method authentication of ESMTP shall be performed
username	Pointer to username string for authentication, ignored if do_auth is 0
password	Pointer to password string for authentication, ignored if do_auth is 0
send_file	1=send file attachment, 0=no file attachment
filename	filename of the file to send, valid only if send_file = 1
callback	Pointer to callback function in user code to receive completion and error event notifications from HS SMTP Library. (See section 3 for more details)
session_ref	User data associated with this session. This parameter is passed to the callback function in user code without modification
use_latin1	Integer flag which switches on the use of iso-8859-1 encoding. for characters from 0x7f to 0xff use escape sequence ==<hex code>. Set this flag to TRUE when sending file attachments. Set to false when sending mail without attachments.
useSSL	0=Standard SMTP, not secure 1= Run SMTP over SSL / TLS using STARTTLS command method 2= Run SMTP over SSL without STARTTLS, SSL connection establishes immediately on TCP connection up. If you set this to 1 or 2, set srv_port to secure SMTP protocol server port, for example 587 or 465 respectively for GMAIL (smtp.gmail.com)
check_peer_certificate	If useSSL=1, set this to 1 to check peer certificate during SSL negotiation
logging_enabled	Enabled logging of debug events using HS SMTP_USR_EV_LOGININFO event

long *session_handle - HS SMTP sets this variable with SMTP session reference. Use the returned session handle to close a session with HsSmtplibAbortMail if required.

Returns:

- *HS SMTP_RC_OK* – Success, mail transfer initiated
- *HS SMTP_RC_NOTINIT* – HS SMTP Library not initialised
- *HS SMTP_RC_INV_PAR* – Invalid parameters specified
- *HS SMTP_RC_NO_CTX* – No free contexts (Maximum 5 concurrent sessions supported)
- *HS SMTP_RC_NONAME* – Socket layer cannot resolve SMTP server name
- *HS SMTP_RC_TCPCONNFAIL* – Socket layer failed to connect to SMTP server

Remarks:

This function initiates sending an email message to the remote SMTP server according to RFC821. The message contains message header, message text and optionally a binary file attachment.

Fill out `hs_smtp_mail_t` structure members and set pointer to storage for SMTP session handle prior to calling the function. The function is non-blocking and complete asynchronously. The completion of mail transfer operation is reported via event callback function. The pointer to callback function is also specified in `hs_smtp_mail_t` structure.

2.5 HsSmtplibAbortMail

Aborts the specified SMTP session in progress.

Syntax:

```
int HsSmtplibAbortMail(long session_handle);
```

Parameters:

long session_handle - SMTP session handle (returned with HsSmtplibSendMail)

Returns:

- *HS SMTP RC OK* – success
- *HS SMTP RC NOTINIT* – HS SMTP Library not initialised
- *HS SMTP RC INV PAR* – Invalid parameters specified
- *HS SMTP RC NOT_OPEN* – mail session not open

Remarks:

This function is used to abort current mail transfer already in progress. The function returns immediately, but the clean mail session termination completes asynchronously and the result is reported via event callback function.

2.6 HsSmtplibContinue

Instructs the SMTP session to proceed connecting to the SMTP server after verification of the server's SSL certificate.

Syntax:

```
int HsSmtplibContinue(long session_handle);
```

Parameters:

long session_handle - SMTP session handle (returned with HsSmtplibSendMail)

Returns:

- *HS SMTP RC OK* – success
- *HS SMTP RC NOTINIT* – HS SMTP Library not initialised
- *HS SMTP RC INV PAR* – Invalid parameters specified
- *HS SMTP RC NOT_OPEN* – mail session not open

Remarks:

This function is only applicable if SMTP over SSL / TLS has been enabled and server certificate validation has been enabled when calling HsSmtplibSendMail(). This function is used to continue to connect to SMTP server after server SSL certificate validation failed and the user wants to ignore server certificate error and continue with mail session. If SSL server certificate validation fails, HSSMTP shall call event callback function with event `HS SMTP_USR_EV_BADCERT`. The user may then call this function if server certificate is not important.

3 HS SMTP to User Event Callback and Events

3.1 Event Callback Prototype

```
typedef int smtp_callback_t(void *session_ref, int ev, long arg1, long arg2);
```

Parameter	Description
*session_ref	User data, passed unchanged from the call to HsSmtplibSendMail
ev	Event code (see event code table below)
Arg1	Parameter 1 (dependant on event code)
Arg2	Parameter 2 (dependant on event code)

3.2 Event Codes

Event	Arg1	Arg2	Description
HS SMTP_USR_EV_DONE	0	0	Mail transfer complete, mail has been successfully sent to SMTP server and mail session is terminated
HS SMTP_USR_EV_CLOSED	0	0	TCP session closed by remote peer
HS SOCK_EV_CONN_FAILED	0	0	TCP connect attempt failed
HS SMTP_USR_EV_SRVERR	Unsigned char *buffer pointer to server reply string	Int length length of server reply string	Mail session closed because SMTP server returned error. The error string returned by the server is passed in parameters arg1 and arg2
HS SMTP_USR_EV_TIMEDOUT	Unsigned char *error Null terminated error string	0	Mail session closed because current protocol operation timed out. Arg1 parameter has pointer to error string with additional information about the timeout condition.
HS SMTP_USR_EV_PROGRESS	Integer number from 0 to 100 representing percentage of transmitted message text	0	Progress indication, parameter 1 has a number from 1 to 100, representing percentage of message text sent so far.
HS SMTP_USR_EV_PROGRESS_F	Integer number from 0 to 100 representing percentage of file attachment transmitted so far	0	Progress indication, parameter 1 has a number from 1 to 100, representing percentage of of file attachment transmitted so far.

HS SMTP_USR_EV_GETNEXT	Pointer to hs_smtp_mail_t message structure	0	When a message has been successfully transmitted to server, the callback function is called with this primitive. If the user application wishes to send the same message to another recipient, it should set the new *recv_addr in the hs_smtp_mail_t structure and return TRUE. If the user application returns FALSE, HS SMTP will exit the mail session
HS SMTP_USR_EV_BADCERT	Pointer to error string	Length of error string	SSL peer certificate verification is on and certificate received failed validation. User must either disconnect (HsSmtAbortMail) or call continue (HsSmtContinue) to ignore this error Until HsSmtContinue is called HS SMTP remains waiting for user decision.
HS SMTP_USR_EV_LOGININFO	Pointer to debug event string	Length of debug event string	If debug event logging is enabled via loggin_enabled member of hs_smtp_mail_t structure, this event carries a debug event string to be printed in application event log