

HS POP3 Lite Module v1.0 User Manual

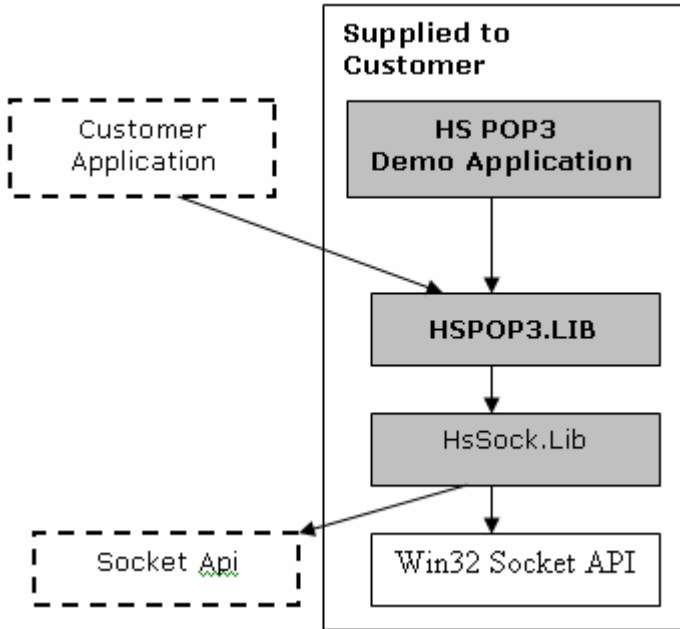
Revision: 1.0
 Date: 27 July 2006

HS POP3 LITE MODULE V1.0 USER MANUAL	1
REVISION: 1.0	1
DATE: 27 JULY 2006	1
1 INTRODUCTION	2
2 HS POP3 API	3
2.1 HSPop3INIT	3
2.1.1 INITIALISATION STRUCTURE DEFINITION (HS_POP3_API_T)	3
2.2 HSPop3DESTROY	4
2.3 HSPop3TICK	4
2.4 HSPop3GETMAIL	5
2.5 HSPop3ABORT	6
2.6 HSPop3GETERRSTR	6
3 HS POP3 TO USER EVENT CALLBACK AND EVENTS	7
3.1 EVENT CALLBACK PROTOTYPE	7
3.2 EVENTS	7
3.2.1 MESSAGE STRUCTURE (HS_POP3_MSG_T)	8

1 Introduction

HS POP3 Lite is a software library in C (supplied with full source code) which implements the client side of Post Office Protocol Version 3 (POP3) over TCP socket layer according to RFC 1939. Among other features, the library supports user authentication, reception of basic internet headers and text, message deletion and statistics.

The library is a stand-alone module which links directly to customer application:



HS POP3 Library incorporates the necessary state machine, transparency procedures, and server response processing required to comply to a simple and robust POP3 client implementation.

2 HS POP3 API

2.1 HsPop3Init

```
int HsPop3Init(hs_pop3_api_t *api);
```

DESCRIPTION

This function initialises HS POP3 Library. It must be called at initialisation, before any other functions are called. Calling HsPop3Init twice will return an error. You can call HsPop3Destroy first to deinitialise HS Pop3 Library and then call HsPop3Init again

PARAMETERS

Type	Name	Description
hs_pop3_api_t	*api	Pointer to parameter structure, please see next section for details

RETURNS

- *HS_POP3_RC_ALRINIT – HS POP3 Library already initialised*
- *HS_POP3_RC_INV_PAR – invalid parameters specified*
- *HS_POP3_RC SOCKINIT_FAIL – socket layer initialisation failed*
- *HS_POP3_RC_OK - success*

2.1.1 Initialisation Structure Definition (hs_pop3_api_t)

Field name	Description
getmem_t *getmem;	<p>Pointer function in user code used by HS POP3 to allocate memory for received messages</p> <p><u>Prototype:</u> unsigned char *getmem_t(int size);</p> <p><u>Parameters:</u> Size – size of memory required</p> <p><u>Returns:</u> pointer to memory allocated or NULL if no memory of that size available</p>
freemem_t *freemem	<p>Pointer function in user code used by HS POP3 to free memory</p> <p><u>Prototype:</u> void freemem_t(unsigned char *p);</p> <p><u>Parameters:</u> p – pointer to memory to free</p> <p>no returns</p>
start_timer_t *start_timer;	<p>Pointer function in user code used by HS POP3 to start a timer</p> <p><u>Prototype:</u> long start_timer_t(long handle, unsigned long secs, timer_callback_t *callback);</p> <p><u>Parameters:</u> Handle – POP3 session handle secs – timeout in seconds *callback – pointer to function in HS POP3 code that the user code should call on timer expiry.</p> <p><u>Callback function prototype:</u></p>

	void timer_callback_t(long handle); handle – HS POP3 session handle <u>Returns:</u> timer handle or NULL if timer start error
stop_timer_t *stop_timer	Pointer function in user code used by HS POP3 to stop a timer <u>Prototype</u> void stop_timer_t(long timer_handle); <u>Parameters:</u> Timer_handle – timer handle (returned from call to start_timer)
event_callback_t *event_cb	Pointer function in user code used by HS POP3 to report operation results, status and errors. (Please see session 3)

2.2 HsPop3Destroy

int HsPop3Destroy(void);		
DESCRIPTION		
<i>This function de-initialises HS POP3 Library, releases all used resources and cleans up Socket Interface Layer.</i>		
PARAMETERS		
Type	Name	Description
None	None	None
RETURNS		
<ul style="list-style-type: none"> • HS_POP3_RC_OK – success, HS POP3 Library successfully de-initialised • HS_POP3_RC_NOTINIT – cannot destroy, HS POP3 Library not yet initialised 		

2.3 HsPop3Tick

int HsPop3Tick (void);		
DESCRIPTION		
<i>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)</i>		
PARAMETERS		
Type	Name	Description
None	None	None
RETURNS		
<ul style="list-style-type: none"> • HS_POP3_RC_OK – success • HS_POP3_RC_NOTINIT – HS POP3 Library not initialised 		

2.4 HsPop3GetMail

```
int HsPop3GetMail(hs_pop3_session_t *s, long *session_handle);
```

DESCRIPTION

This function initiates mail reception from POP3 server. It works as follows:

- 1) HS POP3 contacts POP3 server with login information
- 2) If login is authorised, HS POP3 checks to see if mailbox has any messages
- 3) If mailbox is not empty, HS POP3 follows this procedure for each message until all messages processed:
 - A) Get message id from server and report it to user application
 - B) If user application indicates via return of callback that it wants to receive this message, HS POP3 requests message size from the server and calls getmem callback to allocate necessary memory for the message
 - C) HS POP3 then receives the message from the server, parses out header information and calls event callback function with HS_POP3_USR_EV_DONE_MSG callback passing the pointer to hs_pop3_msg_t structure, which contains various information about the message, parsed out headers, pointer to start of message, message id, message size, size of headers etc. (see hs_pop3_msg_t structure definition for details)
 - D) When all messages are processed, HS POP3 calls event callback with HS_POP3_USR_EV_DONE event passing total number of messages in mailbox and number of received messages

Depending on parameters, HS POP3 can delete messages from the server after successful reception.

PARAMETERS

Type	Name	Description
hs_pop3_session_t	*s	<p>Pointer to session parameters structure:</p> <p>srv_name – POP3 server name (Note 1) srv_ip – POP3 server IP address (Note 1) username – POP3 username, 0 terminated password – POP3 password, 0 terminated srv_port - POP3 server port (usually 110) delete_msgs – is TRUE, delete messages from server user_data – user handle, not modified by HS POP3 and always passed back in event callback function.</p> <p>Note 1: Either server name or server IP address string can be supplied to the function. If srv_ip length is not 0, srv_ip string shall be used to connect to POP3 server. If srv_ip length is 0, srv_name string shall be used to resolve server IP address first before connecting to it.</p>
long	*session_handle	HS POP3 fills in the variable pointed to by this parameter with POP3 session reference. Use this parameter to close a session with HsPop3Abort if required.

RETURNS

- HS_POP3_RC_NOTINIT – HS POP3 Library not initialised
- HS_POP3_RC_INV_PAR – invalid parameters specified
- HS_POP3_RC_NOPASS – no password specified in parameters
- HS_POP3_RC_NOUSER – no username specified in parameters
- HS_POP3_RC_NO_CTX – maximum number of open HS POP3 sessions reached
- HS_POP3_RC_NONAME – POP3 server name could not be resolved
- HS_POP3_RC_TCPCONNFAIL- TCP connection to server could not be established
- HS_POP3_RC_TIMERFAIL – timer start failure
- HS_POP3_RC_OK - success

2.5 HsPop3Abort

Int HsPop3Abort(long session_handle)

DESCRIPTION

This function is used to abort current mail session in progress. HS POP3 library will discard current message (if not fully received) free message memory and exit mail session cleanly via POP3 QUIT command. This means that the session is not closed immediately, but only after reception of a valid response to QUIT command or timeout.

PARAMETERS

Type	Name	Description
long	session_handle	POP3 session handle of the session to abort

RETURNS

- *HS_POP3_RC_OK – success*
- *HS_POP3_RC_NOTINIT – HS POP3 Library not initialised*
- *HS_POP3_RC_INV_PAR – invalid parameters specified*

2.6 HsPop3GetErrStr

unsigned char *HsPop3GetErrStr (int rc)

DESCRIPTION

This function is used to convert integer HS POP3 return code into a readable string – error description

PARAMETERS

Type	Name	Description
Int	rc	HS POP3 integer return code

RETURNS

- *Pointer to zero terminated error string*
- *NULL – error not found (not a valid return code)*

3 HS POP3 to USER Event Callback and Events

3.1 Event Callback Prototype

```
typedef int event_callback_t(long handle, int ev, long arg1, long arg2);
```

Parameter	Description
handle	User handle, the same as specified in call to HsPop3getMail in user_data parameter of hs_pop3_session_t structure
ev	Event code (see next section for list of event codes)
Arg1	Parameter 1, value depends on event
Arg2	Parameter 2, value depends on event

Returns:

True or False. For most events the return is insignificant and is not checked by HS POP3. Where HS POP3 needs a specific return, it is clearly specified in this document

3.2 Events

Event	Arg1	Arg2	Description
HS_POP3_USR_EV_DONE	Long number of messages received	0	Mail session complete
HS_POP3_USR_EV_DONE_MSG	Pointer to message structure hs_pop3_msg_t: (see section 3.2.1)	0	A message successfully received, this event shall be repeated for each message received within a mail session
HS_POP3_USR_EV_CLOSED	0	0	Socket layer closed TCP connection. HS POP3 will release any allocated memory for a message being currently received
HS_POP3_USR_EV_CONNFAIL	0	0	HS POP3 could not connect to POP3 server
HS_POP3_USR_EV_SRVERR	Unsigned char pointer to server reply string	Length of server reply string	Session closed because of POP3 error response received from server HS POP3 will release any allocated memory for a message being currently received
HS_POP3_USR_EV_TIMEDOUT	Unsigned char pointer to additional information string (about the context in which timeout occurred)	Length of additional information string	Timed out waiting for server response, session closed. HS POP3 will release any allocated memory for a message being currently received
HS_POP3_USR_EV_PROGRESS1			HS POP3 reports number of messages in

			mailbox and total size of mailbox in octets. This event is indicated once per POP3 session
HS_POP3_USR_EV_PROGRESS2	Current message number (long)	Number of bytes received so far (long)	<p>current receiving message number and number of bytes received so far.</p> <p>This event is indicated for each message block received until full message is received. This event is indicated for each message within the same session. It can be used to indicated individual message reception progress.</p>

3.2.1 Message structure (hs_pop3_msg_t)

Data Member	Description
unsigned char *msgp;	Pointer to full message including headers as received by HS POP3 in memory block allocated by HS POP3 via callback getmem. From this point it is responsibility of user code to free this memory when the message is no longer needed.
long msglen;	Full message length
int hdrlen;	Length of internet headers. The headers start from the first byte of the message (pointed to by msgp)
unsigned char from[HS_POP3_MAX_PATH];	FROM internet address, parsed out from message header
unsigned char date[HS_POP3_MAX_DATE];	DATE parsed out from message header
unsigned char subj[HS_POP3_MAX_SUBJ];	SUBJECT parsed out from message header
unsigned char msgid[HS_POP3_MAX_MSGID];	Message ID string as received from POP3 server for that message
unsigned char *bodyp;	Pointer to start of message text Text within memory block pointed to by msgp. (message text follows internet headers
long bodylen;	Length of message text within memory block pointed to by msgp.