Application Note #401:

Send Mail API For CH216x iModem Products

OBJECTIVE

This document contains a basic description and detailed instructions concerning Creation and Transmission (i.e., Sending) of Email messages using the CH216x iModem products.

It is assumed that it has been verified that the specific CH216x device has established IP connectivity, and is correctly configured for the Internet Service Provider (ISP).

DOCUMENT CONTENTS

1. OVERVIEW

An SMTP server sends Email to its final destination on behalf of the sender – in this case, the iModem product. Generally, an Email sender must subscribe to an SMTP service with an Email provider that operates SMTP servers. Figure 1A, below, illustrates how the iModem and an SMTP server are connected to the Internet.

Email content is provided from two sources:

A. Files stored on the iModem.
B. Streaming ASCII content supplied one line at a time in real time.

Stored email files may be located in either static or volatile iModem memory. Static files are preserved when the iModem loses power while volatile memory is lost during power down.

Email files stored on the iModem may or may not contain internal Email headers (at the user's discretion). Email headers contain such items as Subject Line, From and To and Email addresses. The iModem may be configured by the user to automatically generate Email headers during the process of Email transmission. For example, the user can configure the iModem for one or more default Email destination addresses.

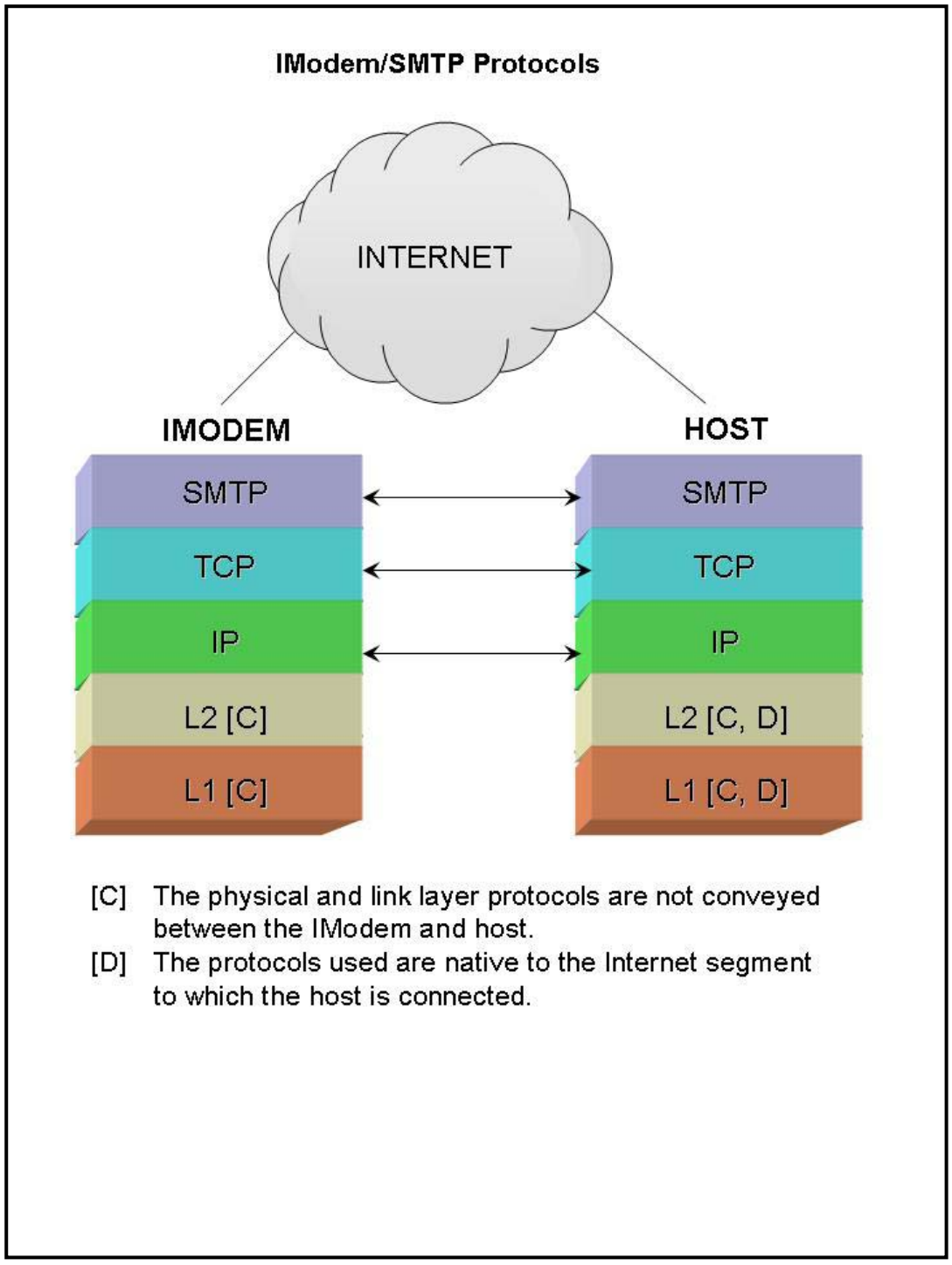

Figure 1A

There are three ways to send Email message using the CH216x iModem products:

A. Using a single API command. Example 1.2 illustrates this method.
B. Using an Email transmission session. Example 1.3 illustrates this method.
C. Using pre-established IP connectivity. Example 1.4 illustrates this method.

The single API command (also referred to as Quick Commands) method is intended primarily for manual transmission of Email messages such as when verifying correct message configuration, to perform experiments or to debug user

applications.

The Email transmission method minimizes the number of API commands needed, and makes extensive use of Email parameters stored in SMTP Email profiles. In addition, this method enables the iModem user to perform other tasks during slow Email operations, including tasks that use other (often simultaneous) TCP connections.

The pre-established IP connectivity method requires the establishment of an IP connection independently of Email operations. It has all the advantages of the Email transmission method while allowing the iModem user to conduct multiple concurrent Email transactions with multiple Email servers. This method also minimizes the overhead consumed during IP connectivity establishment because IP connectivity is maintained until the iModem user explicitly terminates IP connectivity.

IMPORTANT NOTE

Technical Note #101 makes extensive use of Arrowgrams to illustrate the flow of iModem API commands and responses between the user and the CH216x iModem product.

Arrowgrams consist of two columns:

The label USER identifies the left hand column of the arrowgram. This column contains iModem API commands entered by the Hyperterminal/Minicom/iModem user.

The label IMODEM identifies the right hand column of an arrowgram. This column contains API command responses generated by the iModem and displayed by Hyperterminal/Minicom.

Arrows between the right and left columns indicate the direction in which iModem API commands and responses flow. API commands and responses are implicitly terminated by a carriage return <CR>.

The '\' character at the end of either command line or response indicates continuation of that command or response to the next line. This is done to line up the directional arrows on the center of the page for ease of reading.

The string "..." after an iModem response represents a descriptive message appended by the iModem. The iModem is configured to append such messages by default.

Example 1.1: Typical Arrowgram.

USER		**IMODEM**
G_SET_ERROR_MODE: VERBOSE	==>	
	<==	POSTED: Operation Started
	<==	OK: Response mode reset.

The POSTED message indicates the iModem has begun executing the G_SET_ERROR_MODEM command, and the OK message indicates the command completed successfully.
Note that Examples 1.2, 1.3, and 1.4 illustrate alternate methods to generate the same outcome.

Example 1.2: Send Email message using a single (i.e., Quick) command.

USER		**IMODEM**
@QTM: FILE=test	==>	
	<==	POSTED: Operation Started
	<==	OK: Email successfully sent

Example 1.3: Send Email message using an Email Transmission session.

USER		IMODEM
Obtain Email Transmission Session ID 0		
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email transmission Session ID \ allocated
Establish IP connectivity (Use Session ID 0)		
ETX_CONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service ready.
Connect to the SMTP server and send the file		
ETX_SEND_FILE[0]: FILE=test	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email successfully sent
Terminate IP Connectivity		
ETX_DISCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service finished.
Release the Session Activity ID		
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Transmission Session ID \ freed

As a self-test, the user may use Example 1.4, above, to verify proper iModem configuration. If Example 1.4 can not be repeated successfully by the user, then the iModem product is probably not configured properly for Email transmission.

Example 1.4: Send Email using pre-established IP connectivity.

This Example uses two Session Activity IDs as follows:

ID 0: Used for IP connectivity management.
ID 1: Used by the Email application.

USER		IMODEM
Obtain an ISOCKET Session Activity ID 0		
G_GET_SESS_ID: ISOCKET	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: I-Socket Session ID allocated.
Obtain Email Transmission Session ID 1		
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started

	<==	OK[1]: Email transmission ...

Establish IP Connectivity (Use Session ID 0)

IS_IPCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: IP Link is up.

Bind to the Email configuration data (Use Session ID 1)

ETX_CONNECT[1]	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: Email transmission service ready.

Connect to the SMTP server and send the file

ETX_SEND_FILE[1]: FILE=test	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: Email successfully sent

Unbind from the Email configuration data

ETX_DISCONNECT[1]	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: Email transmission service finished.

Terminate IP connectivity

IS_IPRELEASE[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: IP Link is down

Release the Session Activity IDs

G_FREE_SESS_ID[1]	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: Email Transmission ...
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: I-Socket Session ID Freed.

2. SENDING A STORED EMAIL MESSAGE

An Email file stored on the iModem can be sent with automatically generated headers or using headers embedded in the Email file itself. In addition, Email messages can be sent with an attachment. The ETX_SEND_FILE command is used to send Email in all cases.

2.1 Email Without Email Headers.

In the case of an Email message without headers, the iModem file contains only the body of the Email message.

Note that in examples 1.2, 1.3 and 1.4 above, all Email headers were constructed from information contained in the default SMTP profile: smtp.default. However, the resulting Email message has no subject line. The reason is that there is no subject parameter in the SMTP profile. This was a practical decision based on the recognition that it would not be sensible for every Email message to have the same subject line.

It is possible to add a subject and other header information to an Email message by specifying header parameters in the ETX_SEND_FILE command string. Header parameters included in the ETX_SEND_FILE command string have

precedence over the corresponding parameters in the SMTP profile. In Example 2.1, below, all SMTP profile header parameters are overridden by the parameters specifying on the ETX_SEND_FILE command.

Example 2.1: Override stored Email header data.

USER		**IMODEM**
	Obtain Email Transmission Session ID 0	
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email trans...
	Establish IP connectivity (Use Session ID 0)	
ETX_CONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service ready.
	Connect to the SMTP server and send the file	
ETX_SEND_FILE[0]: FILE=test \ PROFILE=smtp.cerz \ SUBJECT="This is a test" \ FROM=dick@best.net \ TO=jane@att.net,spot@dog.net	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email successfully sent
	Terminate IP Connectivity	
ETX_DISCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service finished.
	Release the Session Activity ID	
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[1]: Operation Started
	<==	OK[1]: Email Trans ...

In the event that the ETX_SEND_FILE command fails to properly execute, the iModem would respond with an ERROR message rather than the OK message. Likely failure causes and remedies are addressed in Section 2.4 and in Section 6: Trouble Shooting.

2.2 Email with Internal Headers.

Email messages can be sent with headers which are embedded in a file stored on the iModem. In that case, the headers of an E-Message are exactly those stored in the file with the body. However, each node the message visits on the journey to its destination will still add headers.

IMPORTANT NOTE

The TO, FROM, and SUBJECT parameters can not be specified on the ETX_SEND_FILE command line when an Email message file contains internal headers.

The following is an example of an Email message with internal headers:

Message-ID:<07FC5A0D.631583E9@imodem.net>
From:<joe@imodem.net>
To:<jane@att.net>,<spot@dog.com>
Subject:Internal Addressing Example

This file was created with internal headers.

The INTERNAL parameter must be included to the ETX_SEND_FILE command string to send a message with internal headers as illustrated by Example 2.2, below.

Example 2.2: Use internal header data.

USER		**IMODEM**
Obtain Email Transmission Session ID 0		
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email transmission ...
Establish IP connectivity (Use Session ID 0)		
ETX_CONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service ready.
Connect to the SMTP server and send the file using internal header data		
ETX_SEND_FILE[0]: FILE=test INTERNAL	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email successfully sent
Terminate IP Connectivity		
ETX_DISCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service finished.
Release the Session Activity ID		
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Transmission ...

In the event that the ETX_SEND_FILE command fails to properly execute, the iModem would respond with an ERROR message rather than the OK message. Likely failure causes and remedies are addressed in Section 2.4 and in Section 6: Trouble Shooting.

2.3 Email with an Attachment.

An Email message may be sent with an attachment using the ETX_SEND_FILE command by including the ATTACH parameter and by specifying the name of the file to be attached. Note that the file to be attached must reside on the iModem. Example 2.3 illustrates how to use the ATTACH parameter.

Example 2.3: Add an attached file to an Email message.

USER		IMODEM
Obtain Email Transmission Session ID 0		
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email transmission ...
Establish IP connectivity (Use Session ID 0)		
ETX_CONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service ready
Connect to the SMTP server and send the file and attachment addendum_A		
ETX_SEND_FILE[0]: FILE=test \ PROFILE=smtp.cerz SUBJECT="Attachment Example \ ATTACH=addendum_A	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email successfully sent
Terminate Email Server Connection and IP Connectivity		
ETX_DISCONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission.. service \ finished.
Release the Session Activity ID		
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Transmission ...

In the event that the ETX_SEND_FILE command fails to properly execute, the iModem would respond with an ERROR message rather than the OK message. Likely failure causes and remedies are addressed in Section 2.4 and in Section 6: Trouble Shooting.

2.4 Email Transmission Trouble Shooting.

The ETX_SEND_FILE command can fail for several reasons. If the command fails, the iModem responds to the failure by issuing an ERROR message instead of an OK message. The ERROR message is followed by an error code and a brief message description of the failure cause.

The following error codes and messages correspond to common problems encountered. As can be seen in the examples, the ERROR message is accompanied by a brief explanation often containing a suggestion for solving the problem(s).

ERROR 80: Bad Email file.

No Email file specified or the specified file does not exist.

ERROR 78: No destination address

No Email recipients specified.

ERROR 77: Inconsistent Email addresses.

Specified header information parameters for a message with internal headers.

ERROR 22: Bad Email header file.

Could not create an Email header or attachment--probably due to a memory shortage.

3. SENDING STREAMING EMAIL

Email messages can be sent one line in real time at a time. This method of Email transmission is known as "Streaming Email", and requires establishment of a TCP connection to a SMTP Email server before beginning Streaming Email transmission.

The following commands are used to send Streaming Email messages.

COMMAND	ABBREVIATED FORM
ETX_START_DATA: Makes a TCP connection, performs initial SMTP handshake, and authenticates with the SMTP server if required.	@ETXDSTART
ETX_SEND_DATA: Sends a line of ASCII data.	@ETXDATA
ETX_END_DATA: Terminates Email transmission.	@ETXEND

For more details concerning these commands, see Cermetek document 613-0004, *iModem: Command Reference Manual*.

Example 3 illustrates transmission of a Streaming Email message.

Example 3: Send Streaming Email

Streaming Email messages use little stored header data. Consequently, the TO parameter, which specifies Email recipients, is mandatory parameter of the ETX_START_DATA command string.

USER		IMODEM
Obtain Email Transmission Session ID 0		
G_GET_SESS_ID: TX_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email trans ...
Establish IP connectivity (Use Session ID 0)		
ETX_CONNECT[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email transmission service ready
Connect to the SMTP server and send the message		
ETX_START_DATA[0]: PROFILE=smtp.cerz SUBJECT="Streaming EMail" \ FROM=dick@best.net \ TO=jane@att.net,spot@dog.net	==>	
	<==	POSTED[0]: Operation Started

```
                                        <==     OK[0]: Begin Email Session
```

Send body of the message

```
ETX_SEND_DATA[0]:Dear Jane and Spot,    ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Data Sent

ETX_SEND_DATA[0]:Run quickly !          ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Data Sent

ETX_SEND_DATA[0]:Sincerely, Dick        ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Data Sent
```

Terminate the SMTP server connection

```
ETX_END_DATA[0]                         ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: End of streaming Email data.
```

Terminate IP Connectivity

```
ETX_DISCONNECT[0]                       ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Email transmission service finished.
```

Release the Session Activity ID

```
G_FREE_SESS_ID[0]                       ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Email Trans...
```

In the event a command fails to properly execute, the iModem would respond with an ERROR message rather than the OK message. Likely failure causes and remedies are addressed in Section 6: Trouble Shooting.

ERROR 22: Bad Email header file

Could not create an Email header--probably due to a memory shortage.

4. CREATING STORED EMAIL

Email can be stored on board the IMODEM with API commands. Both Email messages and attachments are stored in iModem files. Those files may reside in volatile memory or static memory. Files in volatile memory are lost when the iModem loses power, and files in static memory are preserved when the iModem loses power.

4.1 Email Messages.

Email messages that contain only a body and no header data can be stored by downloading them directly to the iModem one line at a time. The only parameter option is whether the Email will be stored in static or volatile memory.

The following commands are used to store Email messages.

COMMAND	ABBREVIATED FORM
ES_OPEN_FILE: Opens a file for storing Email.	@ESOPEN
ES_CLOSE_FILE: Closes a file with stored Email	@ESCLOSE
ES_WRITE_DATA: Write a line of ASCII data to an Email file.	@ESWDATA

For more details concerning these commands, see Cermetek document 613-0004, *iModem: Command Reference Manual*.

Example 4.1A illustrates Email storage in static memory and Example 4.1B illustrates Email storage in volatile memory.

Example 4.1A: Store an Email Message in Static Memory.

USER		**IMODEM**
	Allocate a Session ID - 0	
G_GET_SESS_ID: STORE_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email Storage Session ID allocated.
	Create a new Email message file	
ES_OPEN_FILE[0]: FILE=\ message1 CREATE	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email file open
	Write data to the file	
ES_WRITE_DATA[0]:Hi Dick:	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:Have a nice day.	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:Regards, jane	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
	Close and save the file	
ES_CLOSE_FILE[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email file closed
	Release the Session Activity ID	
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Storage Session ID freed.

Example 4.1B: Store an Email message in volatile Memory.

USER		**IMODEM**
	Allocate a Session ID - 0	
G_GET_SESS_ID: STORE_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email Storage Session ID allocated.

Create a new Email message file

```
ES_OPEN_FILE[0]: FILE= \                ==>
message2 CREATE VOLATILE
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Email file open
```

Write data to the file

```
ES_WRITE_DATA[0]:Hi Dick:               ==>
                                        <==  POSTED[0]: Operation Started
                                        <==  OK[0]: Data Stored

ES_WRITE_DATA[0]:Have a nice day.       ==>
                                        <==  POSTED[0]: Operation Started
                                        <==  OK[0]: Data Stored

ES_WRITE_DATA[0]:Regards, Jane          ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Data Stored
```

Close and save the file

```
ES_CLOSE_FILE[0]                        ==>
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Email file closed
```

Release the Session Activity ID

```
G_FREE_SESS_ID[0]                       ==>
                                        <==  POSTED[0]: Operation Started
                                        <==  OK[0]: Email Storage Session ID freed.
```

4.2 EMail Messages with Headers.

Storing an Email message with headers is similar to storing an Email without headers. However, the header data must be specified as part of the ES_OPEN_FILE API command string. Example 4.2 illustrates storage of an Email with internal header data.

Example 4.2: Store Email with header data.

USER **IMODEM**

Allocate a Session ID - 0

```
G_GET_SESS_ID: STORE_MAIL \             ==>
                                        <==     POSTED: Operation Started
                                        <==     OK[0]: Email Storage Session ID allocated.
```

Create a new Email message file

```
ES_OPEN_FILE[0]: FILE= \                ==>
message3 CREATE \
SUBJECT=Greetings  \
FROM=jane@att.net  \
TO=dick@best.net
                                        <==     POSTED[0]: Operation Started
                                        <==     OK[0]: Email file open
```

Write data to the file

USER		IMODEM
ES_WRITE_DATA[0]:Hi Dick:	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:Have a nice day.	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:Best Regards, Jane	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Data Stored
Close and save the file		
ES_CLOSE_FILE[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email file closed
Release the Session Activity ID		
G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Storage Session ID freed.

4.3 Email Attachments.

Storing an Email attachment is accomplished by using the ES_OPEN_ATTACH command to create and store the attachment. Example 4.3 illustrates how to create an Email attachment.

Example 4.3: Create and store an Email attachment.

USER		IMODEM
Allocate a Session ID 0		
G_GET_SESS_ID: STORE_MAIL	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email Storage Session ID allocated.
Create a new Email attachment file		
ES_OPEN_ATTACH[0]: FILE=attachment1 \ CREATE VOLATILE	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email file open
Write data to the file		
ES_WRITE_DATA[0]:The amount	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:of the bill	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Data Stored
ES_WRITE_DATA[0]:is $3000.00	==>	
	<==	POSTED: Operation Started

USER		IMODEM
	<==	OK[0]: Data Stored

Close and save the file

ES_CLOSE_FILE[0]	==>	
	<==	POSTED: Operation Started
	<==	OK[0]: Email file closed

Release the Session Activity ID

G_FREE_SESS_ID[0]	==>	
	<==	POSTED[0]: Operation Started
	<==	OK[0]: Email Storage Session ID freed.

4.4 Email Management Utilities.

The iModem products provide a variety of commands for managing stored Email. These are briefly described below and in detail in Cermetek document 613-0004, *iModem: Command Reference Manual*.

COMMAND	ABBREVIATED FORM
ES_LIST: List Email messages stored on the iModem and their attributes	----------
ES_DELETE: Delete Email messages from the iModem	----------
ES_LIST_ATTACH: List Email attachment stored on the iModem and their attributes	----------
ES_DELETE_ATTACH: Delete Email attachments from the iModem	----------
ES_READ_DATA: Review Email files and attachments.	----------

4.5 Email Storage Trouble Shooting.

If the command fails, the iModem responds to the failure by issuing an ERROR message instead of an OK message. The ERROR message is followed by an error code and a brief message description of the failure cause.
The following error codes and messages correspond to commonly encountered problems. As can be seen in the examples, the ERROR message is accompanied by a brief explanation often containing a suggestion for solving the problem(s).

ERROR 82: Illegal operation on existing file.

Header data parameters can not be added to an existing file.

ERROR 78: No destination address

Header data parameters included but no Email recipients are specified.

ERROR 30: The file already exists.

ERROR 15: Error during file operation.

Can't create the file.

5. CONFIGURING EMAIL PARAMETERS

Most Email parameters can be configured with iModem commands, or by editing a SMTP profile and downloading the modified profile into the iModem with the aid of a Profile Download program. However, a select few parameters can only be configured by editing a SMTP profile. The configurable Email parameters are listed below with modification methods indicated:

PARAMETER	MODIFICATION METHOD
Email Addresses	[D]
The SMTP TCP port number	[A]
The SMTP server	[C]
The TCP Profile	[A]
The Link Profile (i.e., ISP data.)	[A]
The SMTP authentication method	[B]
The SMTP account	[A], [B]
The SMTP Login ID and Password.	[B]
The iModem domain (e.g., imodem.net.)	[A], [B]

Legend:
[A] Only configurable by editing the SMTP profile.
[B] Used only by SMTP servers that require authentication.
[C] Optionally configurable by editing the TCP profile.
[D] Use an iModem command.

An SMTP (Simple Mail Transfer Protocol) server is an Internet host operated by the Email service provider. The SMTP server sends the Email on behalf of the subscriber. There are two kinds of SMTP servers: Authenticated and Unauthenticated. Authenticated SMTP servers require the subscriber to log in with authentication data in order to gain access to the SMTP servers services. Unauthenticated SMTP servers do not require authentication data.

IMPORTANT NOTE

The @SHOW API command can be used to verify the SMTP configuration. It is typically used as follows: @SHOW: PROFILE=smtp.cerz LEVEL=TOP.

5.1 SMTP Configuration using iModem Commands.

The following Email parameters can be configured using the ETX_CONFIG API command (abbreviated form of command: @ETXC). For more details, see Cermetek document 613-0004, *iModem: Command Reference Manual*.

The SMTP profile contains all parameters listed below EXCEPT the SMTP name. The SMTP server name resides within the applicable TCP Profile (e.g., *tcp.smtp.ispname*).

PARAMETER	MODIFICATION METHOD
Email Addresses	------
The SMTP server	[B]
The SMTP authentication method	[A]
The SMTP Login ID and Password	[A]

Legend:
[A] Used only by SMTP servers that require authentication.
[B] This parameter is in the TCP profile.

The "TO" parameter is the destination Email address. It can be used to specify single or multiple recipients. Examples of both cases are shown below. Note that multiple Email addresses are separated by the ',' character with no spaces between Email recipient addresses.

Example 5.1A: Single Email Recipient TO command string.

```
TO=dick@best.net
```

Example 5.1B: Multiple Email Recipients TO command string.

TO=dick@best.net,jane@att.net,spot@dog.com

Both text and IP4 format can be used to specify an SMTP server. The general formats are as follows:

SERVER=TEXT-<Server_Name>	Where **Server_Name** is the name of server (i.e., **smtp.earthlink.net.**).
SERVER=IPV4-<ip_address>	Where **ip_address** is the IPV4 format server Internet address such as **64.227.162.49**.

IMPORTANT NOTE

The PROFILE parameter must specify a valid profile name.

Example 5.1C: Configure the SMTP server and Email addresses.

USER		**IMODEM**
ETX_CONFIG: PROFILE=smtp.cerz \ SERVER=TEXT-smtp.main.cz\ FROM=dick@best.net TO=jane@att.net,spot@dog.net	==>	
	<==	POSTED: Operation Started
	<==	OK: Configuration OK

In the event that the ETX_CONFIG command fails to properly execute, the iModem would respond with an ERROR message rather than the OK message. Likely failure causes and remedies are addressed in Section 5.3: Trouble Shooting.

Some Email providers require subscribers to authenticate before sending Email. Generic protocol SMTP servers used for Email authentication are known as SASL and are described in RFC 2554. When authentication is required, subscribers usually have no choice but to use the mechanism required by their Email provider. Common authentication methods are briefly summarized below:

- CRAM-MD5: Authentication data is encrypted, and exchanged via the challenge mechanism described by RFCs 4422 and 1321. This method is generally considered secure.
- PLAIN: Authentication data is sent as base 64 encrypted strings. This method is generally not considered secure.
- LOGIN: Authentication data is sent as unencrypted ASCII strings. This method is generally not considered secure.

The ETX_CONFIG command is used to configure the authentication mechanism and provide the required data for authentication. The ETX_CONFIG command and associated parameters are briefly described below.

AUTH_METHOD	Used to specify Authentication Method. Allowed values:	
	CRAM_MD5	Configures the iModem to use the method corresponding to this name.
	PLAIN	Configures the iModem to use the method corresponding to this name.
	LOGIN	Configures the iModem to use the method corresponding to this name.
	ANY	Configures the IMODEM to automatically negotiate best authentication method with the SMTP server.
	NONE.	Configures the iModem not to use authentication.
ID	The SMTP Login ID. This parameter should not be used if AUTH_METHOD is set to NONE.	
PWD	The SMTP password. This parameter should not be used if AUTH_METHOD is set to NONE.	

Example 5.1B: Configure automatic SMTP authentication negotiation.

USER		**IMODEM**
ETX_CONFIG: PROFILE=smtp.cerz \ AUTH_METHOD=ANY \ ID=me PWD=guess	==>	
	<==	POSTED: Operation Started
	<==	OK: Configuration OK

The iModem will respond with an ERROR rather than OK message if the ETX_CONFIG command fails to properly execute. Section 5.3: Trouble Shooting describes typical failure causes and recommends possible solutions.

5.2 Configuration with a SMTP Profile.

An SMTP Profile can be edited on a PC using any available ASCII editor (e.g., NotePad, WordPad, etc.). Cermetek recommends selecting an SMTP Profile similar in configuration to the requirements of the user's selected Email service provider and modify that SMTP Profile to create an SMTP Profile for the user's EMAIL service provider. Cermetek has working SMTP profiles for a variety of Email providers.

The SMTP Profile is divided into three subset profiles containing associated of parameters. Each subset profile has the suffix <PROVIDER> attached to the profile's name. For the case of the CH2166 Analog iModem, <PROVIDER> is the name of the ISP through which Email service is provided. For the case of the CH2168 GPRS/RF iModem, <PROVIDER> is the name of the Email provider. A typical example: smtp.sbc.

Application Profile: This file contains the TCP port number parameter, links to the other parts of the profile, links to the TCP and Link profiles. The Application Profile generally does not require reconfiguration for a specific user Email service provider. See Appendix A for an Application Profile example. The Application Profile file name has the following form:

smtp.<PROVIDER> The default file name is smtp.default.

Authentication Profile: This file contains Email addresses and a link to the SMTP Account Profile. The Authentication Profile generally applies to a single Email provider account. See Appendix A for an example. The Authentication Profile file name has the following form:

auth.smtp.<PROVIDER> The default file name is auth.smtp.default.

Account Profile: This file contains the necessary authentication data for SMTP servers that require authentication. The Account Profile applies to a single Email account. See Appendix A for an example. The Account Profile file name has the following form:

acct.smtp.<PROVIDER> The default file name is acct.smtp.default.

IMPORTANT NOTE

The SERVER_ADD and SERVER_FMT_ADD field of the TCP profile are used to configure the SMTP server. The SERVER_FMT_ADD field is the server address format and can be either TEXT or IPV4. The SERVER field specifies the server and can be a Symbolic Name or an IPV4 address.

Example 5.2A: Configure an IPV4 server address.

SERVER_FMT_ADD = IPV4
SERVER_ADD = 66.117.140.246

The next few sections summarize the most commonly used SMTP profile fields.

5.2.1 Application Profile Field Summary.

TCP_PORT_NO The SMTP port number used by the SMTP server. Typically, 25 is port the used but some servers use 587.

AUTH_PROFILE A logical link to Authentication Profile. This field contains the actual name of the Authentication Profile.

ACCT_PROFILE A logical link to the Account Profile. This field contains the name of the Account Profile.

TCP_PROFILE A logical link to the TCP Profile. the name of the TCP Profile.

LINK_PROFILE A link--the name of the Link Profile.

5.2.2Authentication Profile Field Summary.

SOURCE_ADD The source Email address.

DESTINATION_ADD The list of destination Email addresses which must contain 1 or more Email addresses. It has the following format:

```
DESTINATION ADDRESS = LIST
START
<LIST>
END
```

Where LIST is a list of Email Destination addresses. Each address must be on a separate line.

Example 5.2B: DESTINATION_ADD file with multiple Email Destination addresses.

```
DESTINATION ADDRESS = LIST
START
dick@best.net
jane@att.net
spot@dog.com
END
```

ACCOUNT: The authentication account name. It must match the account field in the Account Profile, and provides a link to the SMTP authentication data.

5.2.3 Account Profile Field Summary.

ACCOUNT The account name which must match the ACCOUNT field of the Authentication Profile.

AUTH The SMTP authentication method. If it is used, it specifies an authentication method. The following values are valid:
CRAM-MD5
LOGIN
PLAIN

If the Authentication field is omitted or commented out, the Authentication process proceeds in one of two ways. First, if a USER and PASSWORD is specified, the IMODEM and SMTP server automatically negotiates the authentication method. Otherwise, authentication is not used.

USER The Login ID. It is only used for SMTP servers that require authentication. Otherwise, it should be omitted or commented out.

PASSWORD The Password. It is only used for SMTP servers that require authentication. Otherwise, it should be omitted or commented out.

DOMAIN The Email domain of the iModem.

HOST The SMTP server name. It should match the SERVER_ADD field in the TCP Profile being used.

PORT The TCP port number for SMTP. It should match the TCP_PORT_NO of the TCP Profile being used. It is typically 25 but occasionally is 587.

5.2.4Downloading a SMTP Profile.
When the ASCII SMTP Profile is ready for downloading into the iModem, use the Profile Download Program to download the SMTP Profile into the iModem. See Application Note iModem.612-0409, iModem Profile Download Tool, for details. Example 5.2.4illustrates how to download an ASCII SMTP Profile file.

IMPORTANT NOTE

Replace "cerz" with "default" to download the default Link Profile or to the name of your service provider to download that provider's profile. Also, replace 2 with the number of the PC serial port in use.

Example 5.2.4: Download an ASCII SMTP Profile from a PC into the iModem. This sequence assumes that the LOAD program resides on the user's PC.

```
load -d2 -p -f smtp.cerz smtp.cerz
load -d2 -p -f acct.smtp.cerz acct.smtp.cerz
load -d2 -p -f auth.smtp.cerz auth.smtp.cerz
load -d2 -p -f tcp.smtp.cerz tcp.smtp.cerz
```

The command flags used in Example 5.2.4 have the following significance:

-d2 Use serial port 2 on your PC.
-p Indicates a normal profile file.
-f Overwrite the old version of the profile if it exists.

If the following message is not the last one displayed by load, then downloading failed.

Load END: SUCCESS

Some typical failure problems are as follows:
If downloading failed, the string "END:" will be followed by a brief message describing the cause. Some common causes are:

- A bad file name was specified.
- The wrong port number was specified.
- The iModem is not completely initialized.
- Another program, HyperTerminal in particular, has the port open.

5.3 Configuration Trouble Shooting.
The ETX_CONFIG command can fail for several reasons. The iModem will respond with an ERROR rather than OK message if it fails, and the ERROR message will be followed by an error code and brief message describing the cause.

The following error codes and messages correspond to common problems. They are accompanied by brief explanations and suggestions for solving the problems.

ERROR 31: No profile or bad profile name was specified.

ERROR 56: The authentication data is inconsistent. For example, no SMTP authentication method was specified but, a Login ID or password was specified.

ERROR 59: The SERVER parameter format is wrong or there is some other parameter format problem.

ERROR 75: An invalid AUTH_METHOD value was specified or some other parameter has a bad value.

6. GENERIC EMAIL TROOUBLE SHOOTING

The following error codes and messages apply to all methods of sending Email, and correspond to common problems. They are accompanied by brief explanations and suggestions for solving the problems.

ERROR 32: Bad Server

This error can occur for two reasons. First, a symbolic SMTP server name was specified and DNS address resolution failed. Second, an invalid SMTP server IP address was used. It means the default iModem SMTP Email server is incorrectly configured, or there is DNS server problem.

ERROR 38: TCP Connection Failure

This error indicates that a TCP connection attempt, to the SMTP server, failed and may occur for several reasons. Among them are the following:

- The TCP server is down.
- An Internet segment between the IMODEM and server is down.
- The Server refused the connection for security reasons.
- A timeout occurred before the connection completed due to network congestion, or slow server response.

The connection may succeed at a later time in this case.

ERROR 56: Bad Login ID or Password

The iModem supplied incorrect authentication data to a SMTP server that requires authentication. The default iModem SMTP Email authentication data is not configured correctly.

ERROR 57: SMTP rejected mail.

An authentication error, like that in error 56 above, occurred, or the SMTP Server refused to send the Email because of its security policies, forwarding policies, spam filters etc.

ERROR 14: Error during SMTP message transmission.

The SMTP server could not send the Email message, or Email transmission was unexpectedly aborted by the SMTP server or network. In this case, Email transmission may work a later time.

APPENDIX A
SMTP Profile Examples

Lines that begin with the '#' character are comments. The syntax of most fields is as follows:

<Field_Name> = <Value>

Example A1. SMTP Application Profile.

```
#
# Default SBC SMTP APPLICATION profile.
#

#
# The application type.
#
TYPE          =     SMTP
#
# The TCP port. No. for the application. (i.e. Usually 25 )
#
TCP_PORT_NO    =     587
#
# Use a dialup IP link.
#
DIALUP        =     TRUE
#
# The name of the APPLICATION SMTP/AUTHENTICATION Profile.
#
AUTH_PROFILE   =     auth.smtp.sbc
#
# The name of the INTERNET/TCP Profile.
#
TCP_PROFILE    =     tcp.smtp.sbc
#
# The name of the IP PROTOCOL FILE Profile.
#
IP_PROFILE     =     ip.dialup_default
#
# The name of the LINK Profile which is only used for dialup IP links.
# IMPORTANT: Only used if DIALUP above is TRUE.
#
ISP_PROFILE    =     link.kore
```

Example A2. SMTP Authentication Profile

```
#
# Default SBC SMTP/AUTHENTICATION profile.
#

#
# The user account name. This must match the name in acct.smtp.sbc
#
USER_ACCOUNT      =   default
#
# SMTP Account Profile name
#
```

```
ACCOUNT_PROFILE   =   acct.smtp.sbc
#
# SMTP Email source address.
#
SOURCE_ADD        =   user100@sbcglobal.net
#
# SMTP Email destination addresses.
# Note: It is possible to specify multiple destination addresses.
#
DESTINATION_ADD   =   LIST
START
  paulr@rcom-software.com
  hroskos@cermetek.com
END
```

Example A3. SMTP Account Profile

```
#
# Default SBC SMTP ACCOUNT profile.
#
#
#
# The account name value must match the value in the USER_ACCOUNT
# field of auth.smtp.sbc--the SMTP Authentication profile.
#
account      default
#
# Authentication Method: ANY
#
# The authentication method is automatically chosen since none is specified.
# The "user" and "password" must be specified correctly.
#
# auth cram-md5
#

#
# SMTP Mail server: Both symbolic and IPV4 format work.  It should match
# the SERVER_ADD field of tcp.smtp.sbc--  the SMTP Application TCP profile.
#
host      smtp.att.yahoo.com
#
# SMTP/TCP Port number: Default SMTP port 25.  It should match the
# TCP_PORT_NO field in smtp.sbc--the SMTP Application profile.
#
port        587
#
# Local system Email originator domain: The domain sent in the SMTP EHLO
# command. It should be the same as the SRC_ADDRESS field in acct.smtp.sbc--
# the SMTP Account profile.
#
domain      sbcglobal.net
#
# SMTP Authentication Login ID.
#
user        user100@sbcglobal.net
#
# SMTP Authentication Password.
#
```

```
password    x100y200
#
# TLS encryption is enabled if this is specified.  Some implementations
# require that the  SMTP STARTTLS command  is disabled in order to use
# TLS encryption.
#
# tls

#
# Disables SMTP STARTTLS command if specified.  It should  be specified
# only if the TLS is being used.
#
# tls_nostarttls
```

APPENDIX B
References

Cermetek Microelectronics Documents

612-0400 Application Note # 400: iSocket API For CH216x iModem Products
612-0401 Application Note # 401: Send Mail API For CH216X iModem Products
612-0404 Application Note # 404: CH2166 Analog iModem Network Configuration Guide
612-0407 Application Note # 407: CH2166 Analog iModem ISP/PSTN Configuration Guide
612-0409 Application Note # 409: iModem Profile Download Tool
613-0004 IMODEM: Command Reference Manual

Internet RFCs

RFC 791: Internet Protocol(IP)
RFC 793: Transmission Control Protocol(TCP)
RFC 1321: The MD5 Message-Digest Algorithm
RFC 2554: SMTP Service Extension for Authentication
RFC 2821: Simple Mail Transfer Protocol(SMTP)
RFC 4422: Simple Authentication and Security Layer.(SASL)
RFC 4954: SMTP Service Extension for Authentication