Generating email from mainframe

Creating internet email from mainframe systems is relatively easy. Understanding how it works, however, requires the user to peel off different layers of technology for examination. With each layer removed, older technology can be seen underneath. The process is similar to digging through layers of soil for the remains of earlier civilizations. Not being a certified expert in this material, I reserve the right be incorrect in any fact that I present here. Still, this background should help you to understand and implement email on the mainframe.

As I present the information here, I will first introduce the inner layers of the oldest technology. Then I will wrap each layer with newer technologies and newer functions.

Layer 1

At the core is an old communication service called RSCS, Remote Spooling Communication Subsystem. IBM is the company that created this technology in the late 1960's. It was/is used to communicate with various systems: hardware devices like printers, jcl spoolers, or RSCS systems on other mainframes.

This happened way back before anyone had ever heard of the internet. People were soon ready to communicate online together. So they used existing RSCS technology to create new links to universities worldwide in the planet's first public "global networking system". Yale University and the City University of New York were the prime movers in arranging this new network in 1981. They had financial help in the form of a grant from (coincidentally) IBM. They called the network BITNet, the "Because It's Time, Network". BITNet included over 1000 sites in 49 countries, but membership started declining in 1993 when people began realizing the potential impact of the internet as a communications medium.

Most old mainframes should still have an RSCS program active on their system. Getting your own program to talk with RSCS could work differently on each system. Here at BI, it's simply a matter of including a line of JCL. It looks very similar to jcl for printing to remote printers. That's because RSCS still routes both kinds of communications on our mainframe.

```
$$ LST LST=SYS007,DEST=(RSCSBI1,EMAIL)
```

I have experimented and discovered that this method can successfully send data records up to about 180 bytes in length. Going beyond this limit risks having your data records truncated.

Layer 2

Once you are sending lines of text to RSCS, you need to tell RSCS what to do with your data. Those directives use a standard format. This format is called SMTP, and it was published back in 1982. It is described in a document named RFC821: Simple Mail Transport Protocol. Only a few lines of text are necessary to deliver your data to any internet email address. The four basic commands are HELO, MAIL, RCPT, and DATA.

HELO tells RSCS where to open a connection with a receiving system. MAIL is used to identify yourself, or the sending program, to the email interpreter. RCPT directs the mail to a destination. DATA indicates where the actual email text begins. Each command should be put at the beginning of a new text line. They are used at BI as follows:

```
HELO [deleted].COM
MAIL FROM:<WALKERT@BIPERF.COM>
RCPT TO:<WALKERT@BIPERF.COM>
DATA
… data lines go here …
```

A technician by the name of [deleted] initially worked on this system. So he probably took the liberty of naming the email interpreter address after himself. I think that's why it's "[deleted]". The "From" and "To" addresses are simply a mainframe userid followed by "@BIPERF.COM".

These four lines of text should be at the beginning of all text output that is directed to the RSCS system.

Layer 3

The above text is sufficient for delivering email to any recipient from the mainframe. You do not need any additional formatting. Email programs usually display email with extra "header" information in them, though. You need to provide that data so that the end users receive email that "looks normal" to them. This information includes the date that the email was sent, the name of the sender, whom to reply to, subject, etc.

Many of these fields were introduced in the early days of internet development. Back in the beginning, the internet had a different name. It was a military network called ARPANet. So the document that explains these fields is called RFC822: Standard for ARPA Internet Text Messages. This standard was also published in 1982. ARPA is the Advanced Research Projects Agency. This ARPANet went through a few beaurocratic and technical transformations before becoming the backbone of today's internet.

I call these new headers the ARPA fields. Including these fields in your text data is easy. Just write them after the SMTP's DATA statement but before your actual email text. These fields should be written after the DATA statement described above in Layer 2, but still before the actual text data that you want to appear in your email. Each header should be written at the beginning of a new text line. Sample headers include the following:

```
From: Travel Costing online system
Sender: Travel Costing online reports
Reply-To: Terry Walker <WALKERT@BIPERF.COM>
To: Travel Costing user <WALKERT@BIPERF.COM>
Subject: Report – BTC701P
Date: Thu, 02 Sep 1999 09:34:48 -0500
```

The programmer's name and userid should be written to the Reply-To field. It is also a good idea to include both descriptive text and a program name in the Subject field. Don't forget to put the Date in the format listed. Be sure to include the "-0500" field at the end. It tells the system how far off that "9:34am" time was from Greenwich time. Here in the Central timezone, we are five hours off.

Layer 4

If you want to try something fancy, however, like including your text as a file attachment rather than as the actual body of the message, then you need to include even newer technology code in your text.

Multimedia has become popular, and people need easy ways of sending various files back and forth via email. So a new email technology has been introduced to meet these needs. RFC2045: Multipurpose Internet Mail Extension protocol was published in 1996. The short name for it is simply MIME. To introduce MIME into your email, you just include a few more lines of descriptive text into your data (just like we've been doing), and email systems will know how to interpret your email in new and interesting ways.

Before you include these new commands, you need to decide on a specific text "codeword" to use in your output. This should be a plain English keyword or a cryptic alphanumeric string which should **never** occur normally in the beginning of any record line. MIME needs this codeword so that it recognizes different sections of your document correctly. You shouldn't have extra occurrences of this codeword in your email body, or MIME might get confused about how to deliver your document. The word that I use in Travel Costing is "simpleboundary".

As the last line of your header text (after the other SMTP headers), include these two lines:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; "simpleboundary"
```

Be sure to include the codeword in double-quote characters. Include at least one blank line after "Content-Type".

Now you are ready to write the different sections of your email. You can write anything you want to the output stream at this point. It will all be ignored. This is the "Preamble" section of the email. It is generally used just for documentation purposes.

When you are ready to create the section that will appear as the main body of your email message, write a blank line, the following line, and then one more blank line:

```
--simpleboundary
```

Note that this boundary marker for MIME is just your codeword preceded by two dashes. Whatever text you write after this will appear in the main portion of your email message. I write helpful messages to the end user in this area.

```
Your requested Travel Costing online report has completed
its batch processing.  Program results are located in the
attached file.
```

When you are finished with this section and are ready to create the "file attachment" section, you will need to write some more MIME headers. Include a blank line before and after this group of headers. These headers are listed below:

```
--simpleboundary
Content-type: text/plain; charset=us-ascii
Content-Transfer-Encoding: quoted-printable
Content-Description: Travel Costing online report
Content-Disposition: attachment; "report.doc"
```

Again, be sure to include the double-quote character in your output. Write all of your normal report lines after this point. You are done with the required header information. Your end user should receive the email message with the report as a file attachment named “report.doc”.

To complete the MIME document, you should write one more record after all other records have been output. Simple use the following line to close your MIME document:

```
--simpleboundary--
```

This footer record is just the codeword with two dashes both before and after it.

(For trivia buffs, it is true that RSCS formally requires a period “ . ” to close the transmission. But I have encountered errors at BI when using that formality. The above instructions are sufficient for email transmission at BI.)

Sample email transmission document written to RSCS:

```
rec
#      Data written to output device

1      HELO [deleted].COM
2      MAIL FROM:<WALKERT@BIPERF.COM>
3      RCPT TO:<WALKERT@BIPERF.COM>
4      DATA
5
6      From: Travel Costing online system
7      Sender: Travel Costing online reports
8      Reply-To: Terry Walker <WALKERT@BIPERF.COM>
9      To: Travel Costing user <WALKERT@BIPERF.COM>
10     Subject: Report – BTC701P
11     Date: Thu, 02 Sep 1999 14:30:48 –0500
12     MIME-Version: 1.0
13     Content-Type: multipart/mixed; “simpleboundary”
14
15     Preamble goes here.  This text should not appear
16     anywhere in the final email document.
17
17     --simpleboundary
18
15     Your requested Travel Costing online report has completed
16     its batch processing.  Program results are located in the
17     attached file.
18
19     --simpleboundary
20     Content-type: text/plain; charset=us-ascii
21     Content-Transfer-Encoding: quoted-printable
22     Content-Description: Travel Costing online report
23     Content-Disposition: attachment; “report.doc”
24

...    (normal report records go here)

x-2    (blank line)
x-1    --simpleboundary—-
x      (blank line)
```