                    A Mail Box Protocol, Version-2

INTRODUCTION

    Initial reaction to RFC 196, "A Mail Box Protocol", NIC (7141,)
    indicates general agreement on the need for such a mechanism. The
    conventions suggested in RFC 196 assumed only the use of the Data
    Transfer Protocol (in NIC 7104) in order to simplify an initial
    implementation. The valid argument, we believe, has been made that
    sites will also implement the File Transfer Protocol and that as much
    as possible the Mail Box Protocol should be a subset of it. This
    version is in answer to this suggestion.

    The purpose of a mail box protocol is to provide at each site a
    standard mechanism to receive sequential files for immediate or
    deferred printing or other uses. The files for deferred printing
    would probably be stored in intermediate disk files, although details
    of how a file is handled, stored, manipulated, or printed at a site
    are not the concern of this protocol.

    A mail box, as we see it, is simply a write only (from the Network)
    sequential file to which messages and documents are appended,
    separated by an appropriate site dependent code.

    It is also assumed that there would be a program at the sending site
    which sends the file in the format given below with the optional
    control codes when appropriate. This program could probably be
    accessed as a subcommand of the Telnet program.

    The motivation for developing this protocol is the Network
    Information Center's (NIC) need to be able to deliver messages and
    documents to remote sites, and to be able to receive documents for
    cataloging, redistribution, and other purposes from remote sites
    without having to know the details of path name conventions and file
    system commands at each site. Multiple mail boxes (256) are allowed
    at each site and are identified as described below. The default is
    mail box number 0 for use with the standard mail printer defined
    below.

    The only place where the Mail Box Protocol has a potential conflict
    with the File Transfer Protocol is in file naming conventions. The
    File Transfer Protocol assumes that the using site will use a
    filename which follows the access and file path name conventions of

the serving site and that this information would be supplied by the
user. In the Mail Box protocol we would like not to have to
explicitly know the path name conventions at each site.

In other words there is a need for a network virtual pathname
convention. We did not want to solve this problem in general at this
time and in RFC 196, NIC 7141, proposed the use of a separate socket
for mail type delivery and the use of an integer 0-127 to specify the
address of a specific file (Mail Box) to be appended to as the
simplest form of network-wide standard file name convention for an
initial implementation.

To follow more closely the spirit of the File Transfer Protocol, I
would now recommend the Append Request be specifically used and that
the standard socket agreed on for use with the File Transfer Protocol
also be used. Following the byte indicating an Append request, there
would be a standard agreed-upon string of letters followed by a
number, indicating that this is a mail box append request. A
suggested name string would be NETMAIL#, where # is a byte
interpreted as a mail box number 0-255. If the above suggested Mail
Box file naming convention is unsuitable and some other network-wide
standard mail box naming can be agreed on, then it can be used.
Please let me know how you feel about this naming convention.

Given agreement on a standard mail box pathname, then the Mail Box
Protocol can utilize a subset of the File Transfer Protocol
conventions to be given below.

The other problem which was raised about the Mail Box Protocol was
the possibility of someone accidentally or deliberately flooding the
printer of a site with garbage, as there are no access or file size
controls. Some thinking and discussions of this problem have yielded
no simple satisfactory solutions. I would recommend initial
implementations without standard special safeguards in this area.
Safeguards would be a site-dependent option. Standard safeguards for
the above problem can be easily added later if they really prove
necessary and satisfactory ones can be agreed on.

MAIL BOX PROTOCOL - VERSION 2

The Mail Box Protocol will use established network conventions,
specifically the Network Control Program, Initial Connection
Protocol, Data Transfer Protocol, and File Transfer Protocol (as
described in Current Network Protocols, NIC 7104).

The normal transmission for Mail Box 0 is to be Network ASCII. The
standard receiving mail printer for mail box number 0 is assumed to
have a print line 72 characters wide, and a page of 66 lines. The
new line convention will be carriage return (Hex '0D'), (Octal '015')
followed by line feed (Hex '0A') (Octal '012') as per the Telnet
Protocol, RFC 158, NIC 6768. The standard printer will accept form
feed (Hex '0C') (Octal '014') as meaning move paper to the top of a
new page.

It is the sender's responsibility to control the length of the print
line and page. If more than 72 characters per line are sent, or if
more than 66 lines are sent without a form feed, then the receiving
site can handle these situations as appropriate for them. These
conventions can be changed by control codes as described below.

At the head of the message or document sent to mail box number 0
there is to be an initial address string terminated by a form feed.
This address string is to contain the sender's name and address, and
the receiver's name and address formatted in some reasonable, easy-
to-read form for a clerk to read and distribute. Comments could also
be included in the address string.

The format of information in mail boxes other than mail box number 0
is not explicitly defined by this protocol.

Initial Connection

Initial Connection will be as per the Official Initial Connection
Protocol, Document #2, NIC 7101, to the standard File Transfer socket
not yet assigned. A candidate socket number, socket #3, has been
suggested.

File Transfer

The mail item (file) to be transferred would be transferred according
to the File Transfer Protocol.

As per the File Transfer Protocol, a file (mail item) can be sent in
more than one data transaction as defined in the Data Transfer
Protocol. End of file is indicated by the file separator (as defined
in Data Transfer Protocol) or by closing the connection.

Order of Transactions

   The only basic operation required is an append.

                       Append Request

   (Mailer) User -------------------> server (Mail Box)

                      <File - data>

                   ------------------>

                   End of File indication

                   ------------------>

                       Acknowledge

                   <------------------

   The data type default is network ASCII. The standard line printer
   default is as defined above. Other control transactions can be used.

CONTROL TRANSACTIONS TO BE USED

   OP CODE

   Hex   Octal
   00   000   Change data type identifier
   09   011   Error or unsuccessful terminate
   0A   012   Acknowledge or successful terminate
   0B   013   Append request (add to existing file)
   5A   132   Change printer control settings

DATA TYPE CODES

   All data types of the File Transfer Protocol can be used for special
   applications. For Mail Box 0, default is 8 bit bytes of Network
   ASCII characters.

ERROR CODES

   All error codes defined in the File Transfer Protocol could be
   returned.

PRINTER CONTROL CODES

```
Hex     Octal
01      321     Meaning:  Set line width to 72 characters
02      322     Meaning:  Use the full width of your printer
03      323     Meaning:  Set page size to 66 lines
04      324     Meaning:  Set page size to infinite
```

Other virtual printer control codes can be added in the future.
Other classes of control codes can be added as the need arises.

<JOURNAL>7612.NLS;1, 27-AUG-71 10:41 RWW ; (Expedite) Title:
Author(s): Richard W. Watson/RWW; Distribution: SDC2 TFL JWM JFH REL
AOJO JEW AWH DLM PWF RAW HRVZ AAM RLS JMM JMW AKB PMK TNP ASL BMW JAM
EAF RTB JMP BDW JTM JCL AJB CDS RFH EMA;/NWG; Sub-Collections: NWG
ARC NIC; RFC# 221; Clerk: RWW;
Origin: <WATSON>MAIL.NLS;4, 27-AUG-71 9:51 RWW ;

        [This RFC was put into machine readable form for entry]
          [into the online RFC archives by Ryan Kato 6/01]