                    LETTER TO TIP USERS -- 2

     This is the second letter to TIP users. The first was RFC #365.
There will be more letters to TIP users as they seem to us to be a
good way to keep you informed about what's going on. We suggest you
keep these letters with your TIP User's Guide (TUG) as we will use the
letters to provide documentation of TIP system changes which are made
before we can get TUG revisions printed and distributed. (It is almost
inevitable that the TUG revisions follow actual system changes.
Further- more, these letters will allow us more discussion of new
commands than in TUG.)

     Some of the changes we will be making to the TIP have been
suggested by TIP users. We won't bother with acknowledg- ments.

     The @PROTOCOL TO LOGIN and @PROTOCOL TO CLOSE BOTH commands will
be removed very soon. We presume no one uses these commands any more
since they have been replaced by @LOGIN and @CLOSE.

     As we warned in TIP Letter 1, the @LOGIN command will be given a
parameter soon, the Host number up to now given with the @HOST
command. At the same time, @HOST will be changed so it does a
simultaneous @RECEIVE FROM HOST and @SEND TO HOST. Presently, @HOST
is the same as @SEND TO HOST.

     Several changes will be made to the @TRANSMIT commands very soon.
First @TRANSMIT ON NO CHARACTERS and @TRANSMIT ON EVERY CHARACTER will
be removed. Their functions will be covered by the other @TRANSMIT
commands. @TRANSMIT NOW will continue to function as at present; it
will cause the one message presently being accumulated to be sent as
soon as possible. @TRANSMIT ON LINEFEED and @TRANSMIT ON MESSAGE-END
will continue to cause the message being accumulated to be sent on
linefeed and CONTROL-S. However, they will additionally cause the
message being accumulated to be sent when the character buffer is
almost full. Thus, it will no longer be necessary to give a @TRANSMIT
EVERY <big number> with @TRANSMIT ON LINEFEED and @TRANSMIT ON
MESSAGE-END. @TRANSMIT EVERY # will continue to cause the message
being accumulated to be sent as near as possible to every #th
character. However, values of # which are bigger than the size of the

                                                          [Page 1]
_____

input buffer will cause transmission when the buffer is almost full;
and a value of 0 for # will reset the terminal to its initial setting
-- TRANSMIT-ON-LINEFEED mode off, TRANSMIT ON MESSAGE- END mode off,
and transmitting every character. Thus, TRANSMIT EVERY 0 has the
effect of the removed @TRANSMIT ON NO CHARACTER command, and @TRANSMIT
EVERY 1 has the effect of the removed @TRANSMIT ON EVERY CHARACTER
command.

     There are two ways outside of letters and the telephone to
communicate your suggestions and complaints to us: log into BBN-TENEX
and SNDMSG to WALDEN or use the NIC Journal system to send a message
to DCW3. Dave likes letters best, incidentally.

     We are going to remove the "NEWS" herald from the TIP's HELLO
message. The problem is that we don't know when everybody has read the
latest news so that we can turn off the herald. Therefore, we can't
turn it off. Therefore, it is useless. Check the NEWS every time you
use the TIP. If once the news begins printing you discover you have
already seen it, you can stop it by typing @CLOSE _LF_ (on a 2741 hit
"attention" first).

     A new TIP message will have been added by the time you get this
letter, the message TIP GOING DOWN. This message will be printed on
every TIP terminal shortly before the TIP is taken down for preventive
maintenance, new software releases, etc. (see RFC #381 for further
discussion of this topic). When this message is printed, all TIP users
should cleanly stop what they are doing with a Host. Eventually, this
message will include information on how long until the TIP will go
down, for how long it will be down, and why.

     While we are on the subject of TIP messages, let us mention that
we will be adding a number of new messages which we believe will
remove some of the present confusion about what the TIP is doing.
Unfortunately, we don't have the space to store the message text
strings, so, we will use numbers for the new messages. The format of
these messages will probably be something like M46 for message 46.
Perhaps when the TIPs get more core we can replace the number-messages
by text-messages.

     We are thinking of changing all the TIP LOGIN commands to OPEN
commands which would be more opposite to the CLOSE commands and not so
liable to confusion with Host LOGIN.

     On page 12 of the TUG is a description of how Hosts can send
commands to a TIP terminal. Be warned, if you decide to use this
facility, that we are changing the TIP command language slowly and we
will not be constrained in these changes by the fact that some Hosts
are sending TIP commands. Therefore, if a Host is going to send a

command to a TIP it ought to implement this in a manner that can be
changed easily.

     Some TIP users have been seeing the following problem. They are
communicating with a Host when suddenly they get the message DEAD. If
they try to LOGIN to the Host again they do not get the DEAD message,
but the Host refuses to allow the LOGIN by either doing nothing,
closing, or refusing. The problem was that occasionally the network
partitioned briefly; for instance, one of the two cross-country lines
was down and the other got flaky for a few seconds. If, during a
period when the network is partitioned, a TIP user sends a message to
a Host which cannot be reached, the TIP types DEAD and closes the
connection to the Host. The Host, on the other hand, may not have been
trying to send to the TIP when the network partitioned; in that case
it might not have noticed that the network partitioned and therefore
still thinks it has an open connection to the TIP. When the TIP then
tries to re-LOGIN to the Host, the Host refuses because it already has
an open connection with that particular TIP device.

     Now that we have three independent cross-country paths we do not
expect this problem to occur often, but if it does we see no
short-term solution. We can't just let a CLOSE reset the connection
since the user's immediately preceeding LOGIN destroyed the Host
supplied socket numbers. One can get out of this state by executing
the Host/Host protocol command from the TIP which resets _all_ TIP
users at the given TIP talking to the given Host; but this is a little
gross. What is maybe needed is a Host/Host protocol command to reset
the Host's connections with a particular user (TIP) socket; we will
try to understand the ramifications of such a command and perhaps
undertake promotion of a Host/Host protocol change. In the meantime,
frequently when the above problem happens some other TIP terminal can
still LOGIN to the Host and then halt the hung terminal's job from the
Host side. If it is not possible to get through on another
connection, a telephone call to the Host, asking them to log the job
out, may be necessary. Or, if there is really no other user talking to
the particular Host, the reset command can be executed -- this command
is not documented but we will tell a responsible person at each TIP
site how to execute the command.

     There is a problem related to the above problem which some TIP
users have seen. Occasionally, an IMP crashes somewhere in the network
and takes a packet of a message along with it. Eventually, the source
of the message gets an incomplete transmission message from the
network. When the TIP gets this message, it closes the connection and
calls the destination dead. This is what most other Hosts do also, we
understand. A more reasonable thing to do might be to retransmit the
message or to tell the user and then let him continue; we would like
to do one of these. But before retransmission or letting the user

continue, the TIP and Host's allocate counters must be resynchronized.
However, there is no Host/Host protocol way to synchronize simple
enough for the TIP to use. What may be needed is a simple Host/Host
protocol reset allocate command. We will try to understand this issue
and, again, perhaps undertake promotion of a Host/Host protocol
change.

     The above two problems explain part of the "lost allocates" but
not all. We have now instrumented the TIP program in a manner which we
hope will help us find the rest of the lost allocate problem soon.

     The TIP's logger (opener?) has been causing users some problems.
Upon examination, the problems were seen to originate from basic
design assumptions within the logger which we are working on changing.
In the short term, however, we think that a discussion of what the
logger is doing and how it works will alleviate some of the grief.

     For the user, opening proceeds in three phases. In the first, the
user is queued up waiting to "get" the TIP's logger. In the second,
the user has gotten the TIP's logger and is beginning the login
sequence. In the third, the user has completed the login sequence and
is waiting for the Host to open up the actual data connections. Many
of the problems stem from the fact that _only_one_user_ may be
proceeding through phase 2 at a time. Hence the need for the queue of
phase 1. Any single user may tie up phase 2 for at most about 1
minute. This is the canonical "timeout" in the logger. Notice that
this does not include the times in either the first or third phases.
Thus, the actual delay before you get a "timeout" after you type @L
can be 1 minute, 2 minutes, 3 minutes...depending on how many other
people are having difficulty logging in at the same time. Abort Login
(@A L) does three different things depending on which phase of logging
in the user is in. In phase 2 it resets the timer to be close to
overflowing so that it is responded to with a "timeout" shortly after
the command is given. In phase 3 it does nothing at all, and in phase
1 it merely removes the user (silently) from the logging queue.

     We will, medium term, have the TIP type out something like "YOUR
LOGGER" when you get off the queue and the logger begins trying to
open your connections. This will at least alleviate user uncertainty
as to whether he is in phase 1 or phase 2. Long term, we will
probably make the logging process reentrant so that users will not
interact with one another quite so destructively. In the short term,
here is a small "cookbook" on how to undo a login that seems to not be
working.

     When you have waited as long as you would like to for the login
to take place, you may type "@A L". If the TIP responds with
"TIMEOUT" in a few second and has not typed T OPEN or R OPEN, then you

are aborted and may attempt logging in again. If it types "TIMEOUT"
but has typed out T OPEN or R OPEN then you should type @C and wait
for that to be responded to (You _must_ wait.) If you get no response
at all to @A L, and the TIP has typed that one or the other connection
is open, you should type @C and wait, as above. Finally, if the TIP
makes no response and has not opened any connection, than you are free
to proceed.

     From now on the name of the DEVICE CODE EXECUPORT command will be
DEVICE CODE EXTRA-PADDING, since there are a number of other terminals
which require this feature. The latest to be added to the list is the
DATAPOINT 3300.

       [This RFC was put into machine readable form for entry]
       [into the online RFC archives by BBN Corp. under the   ]
       [direction of Alex McKenzie.   1/97]