                          TN3270 Enhancements

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   This document describes a protocol that more fully supports 3270
   devices than do the existing tn3270 practices. Specifically, it
   defines a method of emulating both the terminal and printer members
   of the 3270 family of devices via Telnet; it provides for the ability
   of a Telnet client to request that it be assigned a specific device-
   name (also referred to as "LU name" or "network name"); finally, it
   adds support for a variety of functions such as the ATTN key, the
   SYSREQ key, and SNA response handling.

   This protocol would be negotiated and implemented under a new Telnet
   Option and would be unrelated to the Telnet 3270 Regime Option as
   defined in RFC 1041 [1].

TABLE OF CONTENTS

1. Introduction

   Currently, support for 3270 terminal emulation over Telnet is
   accomplished by the de facto standard of negotiating three separate
   Telnet Options - Terminal-Type [2], Binary Transmission [3], and End
   of Record [4]. Note that there is no RFC that specifies this
   negotiation as a standard. RFC 1041 attempted to standardize the
   method of negotiating 3270 terminal support by defining the 3270
   Regime Telnet Option. Very few developers and vendors ever
   implemented RFC 1041.

   This document will refer to the existing practice of negotiating
   these three Telnet Options before exchanging the 3270 data stream as
   "traditional tn3270".

   NOTE: Except where otherwise stated, this document does not
   distinguish between Telnet servers that represent SNA devices and
   those that represent non-SNA 3270 devices.

All references in this document to the 3270 data stream, 3270 data
stream commands, orders, structured fields and the like rely on [5].
References to SNA Request and Response Units rely on [6]. References
to SNA versus non-SNA operation rely on [7].

There are several shortcomings in traditional tn3270; among them are
the following:

  - It provides no capability for Telnet clients to emulate the 328x
    class of printers.

  - There is no mechanism by which a Telnet client can request that
    a connection be associated with a given 3270 device-name. This
    can be of importance when a terminal session is being
    established, since many host applications behave differently
    depending on the network name of the terminal. In the case of
    printer emulation, this capability is an absolute necessity
    because a large number of host applications have some method of
    pre-defining printer destinations.

  - The 3270 ATTN and SYSREQ keys are not universally supported.

  - There is no support for the SNA positive/negative response
    process. This is particularly important if printer emulation is
    to function properly, but is also useful for some terminal
    applications. A positive response is used to indicate that
    the previously received data has been successfully processed.
    A negative response indicates some sort of error has occurred
    while processing the previously received data; this could be
    caused by the host application building a 3270 data stream that
    contains an invalid command, or by a mechanical error at the
    client side, among other things.

  - There is no mechanism by which the client can access the SNA
    Bind information. The Bind image contains a detailed
    description of the session between the Telnet server and the
    host application.

  - There is no mechanism by which the server can determine whether
    a client supports 3270 structured fields, or a client can
    request that it receive them.

2. TN3270E Overview

   In order to address these issues, this document proposes a new Telnet
   Option - TN3270E. Telnet clients and servers would be free to
   negotiate support of the TN3270E option or not. If either side does
   not support TN3270E, traditional tn3270 can be used; otherwise, a

sub-negotiation will occur to determine what subset of TN3270E will
be used on the session. It is anticipated that a client or server
capable of both types of 3270 emulation would attempt to negotiate
TN3270E first, and only negotiate traditional tn3270 if the other
side refuses TN3270E.

Once a client and server have agreed to use TN3270E, negotiation of
the TN3270E suboptions can begin. The two major elements of TN3270E
sub-negotiation are:

  - a device-type negotiation that is similar to, but somewhat
    more complicated than, the existing Telnet Terminal-Type Option.

  - the negotiation of a set of supported 3270 functions, such as
    printer data stream type (3270 data stream or SNA Character
    Stream), positive/negative response exchanges, device status
    information, and the passing of BIND information from server to
    client.

Successful negotiation of these two suboptions signals the beginning
of 3270 data stream transmission. In order to support several of the
new functions in TN3270E, each data message must be prefixed by a
header. This header will contain flags and indicators that convey
such things as positive and negative responses and what type of data
follows the header (for example, 3270 data stream, SNA Character
Stream, or device status information).

3. Command Names and Codes

        TN3270E   40
           ASSOCIATE   00
           CONNECT   01
           DEVICE-TYPE   02
           FUNCTIONS   03
           IS   04
           REASON   05
           REJECT   06
           REQUEST   07
           SEND   08

        Reason-codes
           CONN-PARTNER   00
           DEVICE-IN-USE   01
           INV-ASSOCIATE   02
           INV-DEVICE-NAME   03
           INV-DEVICE-TYPE   04
           TYPE-NAME-ERROR   05
           UNKNOWN-ERROR   06

```
      UNSUPPORTED-REQ     07

    Function Names
      BIND-IMAGE          00
      DATA-STREAM-CTL     01
      RESPONSES           02
      SCS-CTL-CODES       03
      SYSREQ              04
```

4. Command Meanings

    IAC WILL TN3270E

       The sender of this command is willing to send TN3270E
       information in subsequent sub-negotiations.

    IAC WON'T TN3270E

       The sender of this command refuses to send TN3270E information.

    IAC DO TN3270E

       The sender of this command is willing to receive TN3270E
       information in subsequent sub-negotiations.

    IAC DON'T TN3270E

       The sender of this command refuses to receive TN3270E
       information.

    Note that while they are not explicitly negotiated, the equivalent of
    the Telnet Binary Transmission Option [3] and the Telnet End of
    Record Option [4] is implied in the negotiation of the TN3270E
    Option. That is, a party to the negotiation that agrees to support
    TN3270E is automatically required to support bi-directional binary
    and EOR transmissions.

    IAC SB TN3270E SEND DEVICE-TYPE IAC SE

       Only the server may send this command. This command is used to
       request that the client transmit a device-type and, optionally,
       device-name information.

    IAC SB TN3270E DEVICE-TYPE REQUEST <device-type>
          [CONNECT | ASSOCIATE <device-name>] IAC SE

       Only the client may send this command. It is used in response
       to the server's SEND DEVICE-TYPE command, as well as to suggest

another device-type after the server has sent a DEVICE-TYPE
REJECT command (see below). This command requests emulation of
a specific 3270 device type and model. The REQUEST command may
optionally include either the CONNECT or the ASSOCIATE command
(but not both). If present, CONNECT and ASSOCIATE must both be
followed by <device-name>. (See the section entitled
"DEVICE-TYPE Negotiation" for more detailed information.)

   IAC SB TN3270E DEVICE-TYPE IS <device-type> CONNECT
        <device-name> IAC SE

   Only the server may send this command. This command is used to
   accept a client's DEVICE-TYPE REQUEST command and to return the
   server-defined device-name.

   IAC SB TN3270E DEVICE-TYPE REJECT REASON <reason-code> IAC SE

   Only the server may send this command. This command is used to
   reject a client's DEVICE-TYPE REQUEST command.

   IAC SB TN3270E FUNCTIONS REQUEST <function-list> IAC SE

   Either side may send this command. This command is used to
   suggest a set of 3270 functions that will be supported on this
   session. It is also sent as an implicit rejection of a previous
   FUNCTIONS REQUEST command sent by the other side (see the
   section entitled "FUNCTIONS Negotiation" for more information).
   Note that when used to reject a FUNCTIONS REQUEST command, the
   function-list must not be identical to that received in the
   previous REQUEST command.

   IAC SB TN3270E FUNCTIONS IS <function-list> IAC SE

   Either side may send this command. This command is sent as a
   response to a FUNCTIONS REQUEST command and implies acceptance
   of the set of functions sent to it in the REQUEST command. Note
   that the list of functions in the FUNCTIONS IS command must
   match the list that was received in the previous FUNCTIONS
   REQUEST command.

5. Default Specification

   WON'T TN3270E

   DON'T TN3270E

   i.e., TN3270E will not be used.

6. Motivation

    See the section entitled "Introduction".

7. TN3270E Sub-negotiation Rules

    All TN3270E commands and parameters are NVT ASCII strings in which
    upper and lower case are considered equivalent.

    Once it has been agreed that TN3270E will be supported, the first
    sub-negotiation must concern the DEVICE-TYPE (and possibly DEVICE-
    NAME) information. Only after that has been successfully negotiated
    can the client and server exchange FUNCTIONS information. Only after
    both DEVICE-TYPE and FUNCTIONS have been successfully negotiated can
    3270 data stream transmission occur.

7.1 DEVICE-TYPE Negotiation

    Device-type (and device-name) negotiation begins when the server
    transmits the DEVICE-TYPE SEND command to the client. The client
    responds with the DEVICE-TYPE REQUEST command, which must include
    a device-type and may include a device-name request.

    Valid device-types are:

      terminals: IBM-3278-2 IBM-3278-2-E (24 row x 80 col display)
                 IBM-3278-3 IBM-3278-3-E (32 row x 80 col display)
                 IBM-3278-4 IBM-3278-4-E (43 row x 80 col display)
                 IBM-3278-5 IBM-3278-5-E (27 row x 132 col display)
                 IBM-DYNAMIC   (no pre-defined display size)

       printers: IBM-3287-1

    Note that the use of '3278' and '3287' is NOT intended to exclude
    any particular device capabilities; they are used here only
    because they are commonly known designations for a terminal and a
    printer member of the 3270 family of devices. The intention is to
    simplify the device-type negotiation (in comparison to traditional
    tn3270) by minimizing the number of possible device-types, and by
    breaking the association of a specific piece of IBM hardware with
    a related set of data stream capabilities. For example,
    negotiation of device-type IBM-3278-2-E does NOT in and of itself
    preclude the use of any of the functions associated with a
    physical 3279 model S2B. A client's ability to support the more
    advanced functions of the 3270 data stream will be indicated not
    by negotiation of an IBM device type and model number, but rather
    by the combination of Read Partition Query and Query Reply.

All of the terminal device-types support a "primary" display size
of 24 rows by 80 columns. The "-3", "-4" and "-5" types each
support an "alternate" display size as noted in the above list.
The IBM-DYNAMIC device-type implies no pre-defined alternate
display size; this value will be passed from the client to host
applications as part of the Query Reply structured field, and it
can represent any display size the client and the host application
can support.

Terminal device-types with the "-E" suffix should only be
negotiated by clients that are willing to support some subset of
the 3270 "extended data stream". This usually includes at a
minimum support for extended colors and highlighting, but may also
include a number of other functions, such as graphics capability,
alternate character sets, and partitions.

Clients that negotiate a terminal device-type with the "-E" suffix
or the DYNAMIC type, as well as those that negotiate a printer
device-type, must be able to accept and respond to a Read
Partition Query command (see the section entitled "3270 Structured
Fields"). This allows the client to indicate to host applications
which subsets of the 3270 extended data stream the client is
willing to support.

In a VTAM/SNA environment, negotiation of IBM-DYNAMIC as the
device-type should result in a Bind in which the Presentation
Services Usage screen field (the eleventh byte in the logmode's
PSERVIC field) is set to 0x03, indicating that the alternate
screen size will be determined by the Query Reply (Usable Area)

7.1.1 Device Pools

   An explanation of the CONNECT and ASSOCIATE commands first
   requires a discussion of the organization of terminal and
   printer device pools that the server maintains and from which
   it selects device-names to assign to session requests. (The
   terms "device-name", "LU name" and "network name" can be
   considered interchangeable in this document.) Also, for the
   purposes of this discussion, the term "generic session request"
   will be used to describe a request for a session by a Telnet
   client (either traditional or TN3270E) that does not include a
   request for a specific device-name. The term "specific session
   request" will be used to describe a request for a session by a
   TN3270E client that includes a request for a specific device-
   name (either via CONNECT or ASSOCIATE).

   As is the case with traditional tn3270, the TN3270E server must
   maintain a set of terminal device-names. A generic request for

a terminal session would result in the server selecting any
available device-name from this pool. The server, however, may
also maintain a separate pool of terminal device-names which
can only be used to satisfy specific terminal session requests.
This is to ensure that a terminal device that has some
significance to host applications (and is therefore likely to
be the target of a specific session request) is not
"accidentally" assigned to a generic request and winds up
associated with a client that has no use for it. Note that the
reverse situation is allowed. That is, a specific terminal
session request could ask for a device-name that happens to be
in the "generic terminal pool".

For each terminal device (in both the "generic" and the
"specific" pools), the TN3270E server could also have defined a
"partner" or "paired" printer device. There should be a
unique, one-to-one mapping between a terminal and its
associated printer. The reasoning behind such a configuration
is to allow for those host applications that produce printed
output bound for a printer whose device-name is determined by
the device-name of the terminal that initiated the print
request. These printer devices can only be assigned to
specific printer session requests that use the ASSOCIATE
command (see below).

In addition, the TN3270E server may also maintain a pool of
printer device-names that are not associated with any terminal.
These printer devices can only be assigned to specific printer
session requests that use the CONNECT command (see below).
This allows for those host applications that generate printed
output bound for a printer whose device-name is determined by
something other than the device-name of the terminal that
initiated the print request (for example, when the userid of
the person signed on to a terminal determines the print
destination).

Finally, it is possible that a pool of printer device-names
could be maintained and used only to satisfy generic requests
for printers.

7.1.2 CONNECT Command

CONNECT is used by the client to request that the server assign
a specific device-name to this Telnet session; it may be used
when requesting either a terminal or a printer session. The
specified device-name must not conflict with the device-type;
e.g., if the client requests DEVICE-TYPE IBM-3287-1 (a printer)
and specifies CONNECT T1000001, but T1000001 is defined at the

host as a terminal, then the server should deny the request.
Further, if the requested device-name is already associated
with some other Telnet session, or if it is not defined to the
server, the server should deny the request.

7.1.3 ASSOCIATE Command

ASSOCIATE can be used by the client only when requesting a
DEVICE-TYPE that represents a printer. The ASSOCIATE command
requests that this session be assigned the device-name of the
printer that is paired with the terminal named in the request.
If the device-type does not represent a printer, or if the
device-name is not that of a terminal, then the server should
deny the request. It is anticipated that the device-name
specified in this request would be one returned by the server
when accepting a previous terminal session request (see the IS
command below). Since no means of authentication has been
provided for, it is possible that the printer paired with the
terminal specified in the ASSOCIATE command has already been
assigned to some other Telnet session; in this case, the server
should deny the request.

7.1.4 Device Selection Rules

To summarize, assume a TN3270E server has the following device
pools defined to it (device-names that begin with a "T" are
terminal devices; those that begin with a "P" are printers):

```
 Generic Terminal Pool                  Specific Terminal Pool
 --------------------                    ----------------------
 TG000001 <--> PTG00001                  TS000001 <--> PTS00001
 TG000002 <--> PTG00002                  TS000002 <--> PTS00002
 TG000003 <--> PTG00003                  TS000003 <--> PTS00003

 Generic Printer Pool                   Specific Printer Pool
 --------------------                    ----------------------
      PG000001                                PS000001
      PG000002                                PS000002
      PG000003                                PS000003
```

Note that the only pool that absolutely must be defined to the
server is the generic terminal pool. The absence of other
pools (or of partner printers for a terminal pool) simply means
that the server is unable to satisfy as wide a variety of
requests as would be possible if all pools were defined to it.

Given the above configuration, the following rules apply:

- a generic terminal request can only be satisfied from the
  generic terminal pool (device-names TG000001 - TG000003).

- a specific terminal request (allowable only via the CONNECT
  command) can be satisfied from either the generic or the
  specific terminal pool, although it is anticipated that the
  majority of such requests would ask for terminals in the
  specific terminal pool (TS000001 - TS000003).

- a generic printer request can only be satisfied from the
  generic printer pool (device-names PG000001 - PG000003).

- a specific printer request may come in one of two forms:

  via ASSOCIATE: the request can only be satisfied using the
                 partner of the specified terminal, which
                 may be in the generic or the specific
                 terminal pool; therefore, devices in the
                 ranges PTG00001 - PTG00003 and PTS00001 -
                 PTS00003 can be used to satisfy the request.

  via CONNECT: the request can be satisfied either from
                 the generic or the specific printer pools
                 (although, as with specific terminal requests,
                 it is likely that most such requests will name
                 printers in the specific printer pool); this
                 request cannot be satisfied with the partner
                 printer of a terminal in either the specific or
                 the generic terminal pools.

7.1.5 Accepting a Request

The server must accept the client's request or deny it as a
whole - it cannot, for example, accept the DEVICE-TYPE request
but deny the CONNECT portion.

If the server wishes to accept the request, it sends back the
DEVICE-TYPE IS command confirming the requested device-type and
the CONNECT command specifying the device-name of the terminal
or printer assigned to this Telnet session. This device-name
may be the one directly requested (via CONNECT) by the client,
the one indirectly requested (via ASSOCIATE) by the client, or
one chosen by the server if the client specified neither
CONNECT nor ASSOCIATE.

7.1.6 REJECT Command

If the server wishes to deny the request, it sends back the
DEVICE-TYPE REJECT command with one of the following reason-
codes:

Reason code name	Explanation
INV-DEVICE-TYPE	The server does not support the requested device-type.
INV-DEVICE-NAME	The device-name specified in the CONNECT or ASSOCIATE command is not known to the server.
DEVICE-IN-USE	The requested device-name is already associated with another Telnet session.
TYPE-NAME-ERROR	The requested device-name is incompatible with the requested device-type (such as terminal/ printer mismatch).
UNSUPPORTED-REQ	The server is unable to satisfy the type of request sent by the client; e.g., a specific terminal or printer was requested but the server does not have such a pool of device-names defined to it, or the ASSOCIATE command was used but no partner printers are defined to the server.
INV-ASSOCIATE	The client used the ASSOCIATE command and either the device-type is not a printer or the device-name is not a terminal.
CONN-PARTNER	The client used the CONNECT command to request a specific printer but the device-name requested is the partner to some terminal.
UNKNOWN-ERROR	Any other error in device type or name processing has occurred.

The process of negotiating a device-type and device-name that
are acceptable to both client and server may entail several
iterations of DEVICE-TYPE REQUEST and DEVICE-TYPE REJECT
commands. The client should make use of the reason-code
specified by the server in any DEVICE-TYPE REJECT command(s) to
minimize the amount of negotiation necessary. For example, if
the client initially requests that it be assigned a specific
terminal device-name via the CONNECT command, and the server
rejects the request with a reason-code of UNSUPPORTED-REQ, the
client should make no further specific terminal requests in the
negotiations. If at any point in the process either side
wishes to "bail out," it can simply send a WON'T (or DON'T)
TN3270E command to the other side. At this point both sides
are free to negotiate other Telnet options (including
traditional tn3270).

7.2 FUNCTIONS Negotiation

Once the DEVICE-TYPE negotiation has successfully completed (i.e,
when the client receives the DEVICE-TYPE IS command), the client
should initiate the FUNCTIONS negotiation by sending the \.
FUNCTIONS REQUEST command to the server. After this initial
REQUEST command, both sides are free to transmit FUNCTIONS REQUEST
and FUNCTIONS IS commands as needed.

7.2.1 Commands

The FUNCTIONS REQUEST command contains a list of the 3270
functions that the sender would like to see supported on this
session. All functions not in the list are to be considered
unsupported. The function-list consists of a string of 2-byte
entries separated from one another by a single space character.
The list is terminated by the IAC code that precedes the SE
command. Functions may appear in any order in the list.

Upon receipt of a FUNCTIONS REQUEST command, the recipient has
two choices:

 - it may respond in the positive (meaning it agrees to support
   all functions in the list, and not to transmit any data
   related to functions not in the list). To do this, it sends
   the FUNCTIONS IS command with the function-list exactly as it
   was received. At this point, FUNCTIONS negotiation has
   successfully completed.

 - it may respond in the negative by sending a FUNCTIONS
   REQUEST command in which the function-list differs from the
   one it received (and not simply in the order of appearance

of functions in the list; at least one function must have
been added to, or removed from, the list).

To avoid endlessly looping, neither party should add to the
function-list it receives any function that it has previously
added and that the other side has removed.

The process of sending FUNCTIONS REQUEST commands back and
forth continues until one side receives a function-list it is
willing to live with. It uses the FUNCTIONS IS command to
accept the list, and, once this command is received by the
other side, all necessary negotiation has been completed. At
this point, 3270 data stream transmission can begin.

Note that it is possible that the function-list agreed to is
null; this is referred to as "basic TN3270E". See the section
entitled "Basic TN3270E" for more information.

7.2.2 List of TN3270E Functions

The following list briefly describes the 3270 functions that
may be negotiated in the function-list:

Function Name	Description
SCS-CTL-CODES	(Printer sessions only). Allows the use of the SNA Character Stream (SCS) and SCS control codes on the session. SCS is used with LU type 1 SNA sessions.
DATA-STREAM-CTL	(Printer sessions only). Allows the use of the standard 3270 data stream. This corresponds to LU type 3 SNA sessions.
RESPONSES	Provides support for positive and negative response handling. Allows the server to reflect to the client any and all definite, exception, and no response requests sent by the host application.
BIND-IMAGE	Allows the server to send the SNA Bind image and Unbind notification to the client.
SYSREQ	Allows the client and server to emulate some (or all, depending on the server) of the functions of the SYSREQ key in an SNA environment.

See the section entitled "Details of Processing TN3270E
Functions" for a more detailed explanation of the meaning and
use of these functions.

8. TN3270E Data Messages

3270 device communications are generally understood to be block
oriented in nature. That is, each partner buffers data until an
entire "message" has been built, at which point the data is sent to
the other side. The "outbound message" (from host to device)
consists of a 3270 command and a series of buffer orders, buffer
addresses, and data, while the "inbound message" contains only buffer
orders, addresses and data. The end of a message is understood to be
the last byte transmitted (note that this discussion disregards SNA
chaining). The Telnet EOR command is used to delimit these natural
blocks of 3270 data within the Telnet data stream.

In TN3270E, each 3270 message must be prefixed with a TN3270E header,
which consists of five bytes and whose format is defined below (see
the section entitled "The TN3270E Message Header").

A "data message" in TN3270E therefore has the following construction:

        <TN3270E Header><data><IAC EOR>

It should be noted that it is possible that, for certain message
types, there is no data portion present. In this case, the TN3270E
data message consists of:

        <TN3270E Header><IAC EOR>

If either side wishes to transmit the decimal value 255 and have it
interpreted as data, it must "double" this byte. In other words, a
single occurrence of decimal 255 will be interpreted by the other
side as an IAC, while two successive bytes containing decimal 255
will be treated as one data byte with a value of decimal 255.

It is strongly recommended that Telnet commands (other than IAC IAC)
should be sent between TN3270E data messages, with no header and no
trailing IAC EOR. If a TN3270E data message containing either IAC IP
(to be interpreted as 3270 Attention) or IAC AO (to be interpreted as
SYSREQ) is received, the receiver should defer processing the command
until the 3270 data has been processed (see the appropriate sections
for discussion of 3270 Attention and SYSREQ). If a TN3270E data
message containing any other IAC-command sequence (other than IAC
IAC) is received, it is implementation dependent when the IAC-command
sequence will be processed, but it must be processed. The receiver
may process it immediately, which in effect causes it to be processed

as if it had been received before the current TN3270E data message,
or the processing may be deferred until after the current TN3270E
data message has been processed. It is because of this ambiguity
that the presence of Telnet commands within a TN3270E data message
(i.e., between the header and the trailing IAC EOR) is not
recommended; neither clients nor servers should send such data.

8.1 The TN3270E Message Header

   As stated earlier, each data message in TN3270E must be prefixed
   by a header, which consists of five bytes and is formatted as
   follows:

```
   ---------------------------------------------------------------
   | DATA-TYPE | REQUEST-FLAG | RESPONSE-FLAG | SEQ-NUMBER   |
   ---------------------------------------------------------------
     1 byte         1 byte         1 byte         2 bytes
```

8.1.1 DATA-TYPE Field

   The DATA-TYPE field indicates how the data portion of the
   message is to be interpreted by the receiver. Possible values
   for the DATA-TYPE field are:

```
   Data-type Name   Code                  Meaning
   --------------   ----   -------------------------------
   3270-DATA        0x00   The data portion of the message
                           contains only the 3270 data stream.

   SCS-DATA         0x01   The data portion of the message
                           contains SNA Character Stream data.

   RESPONSE         0x02   The data portion of the message
                           constitutes device-status information
                           and the RESPONSE-FLAG field indicates
                           whether this is a positive or negative
                           response (see below).

   BIND-IMAGE       0x03   The data portion of the message is
                           the SNA bind image from the session
                           established between the server and the
                           host application.

   UNBIND           0x04   The data portion of the message is
                           an Unbind reason code.

   NVT-DATA         0x05   The data portion of the message is to
                           be interpreted as NVT data.
```

          REQUEST   0x06   There is no data portion present in
                                      the message. Only the REQUEST-FLAG
                                      field has any meaning.

          SSCP-LU-DATA   0x07   The data portion of the message is
                                      data from the SSCP-LU session.

    8.1.2 REQUEST-FLAG Field

       The REQUEST-FLAG field only has meaning when the DATA-TYPE
       field has a value of REQUEST; otherwise, the REQUEST-FLAG field
       must be ignored by the receiver and should be set to 0x00 by
       the sender. Possible values for the REQUEST-FLAG field are:

       Request-Flag Name   Code   Meaning
       -----------------   ----   --------------------------------
       ERR-COND-CLEARED   0x00   The client sends this to the server
                                   when some previously encountered
                                   printer error condition has been
                                   cleared. (See the section entitled
                                   "The RESPONSES Function" below.)

    8.1.3 RESPONSE-FLAG Field

       The RESPONSE-FLAG field only has meaning for certain values of
       the DATA-TYPE field. For DATA-TYPE field values of 3270-DATA
       and SCS-DATA, the RESPONSE-FLAG is an indication of whether or
       not the sender of the data expects to receive a response. In
       this case the possible values of RESPONSE-FLAG are:

       Response-Flag Name Code   Meaning
       ------------------ ----   --------------------------------
       NO-RESPONSE   0x00   The sender does not expect the
                                   receiver to respond either
                                   positively or negatively to this
                                   message. The receiver must
                                   therefore not send any response
                                   to this data-message.

       ERROR-RESPONSE   0x01   The sender only expects the
                                   receiver to respond to this message
                                   if some type of error occurred, in
                                   which case a negative response must
                                   be sent by the receiver.

       ALWAYS-RESPONSE   0x02   The sender expects the receiver to
                                   respond negatively if an error
                                   occurs, or positively if no errors

occur. One or the other must
always be sent by the receiver.

For a DATA-TYPE field value of RESPONSE, the RESPONSE-FLAG is
an actual response to a previous data message (which must by
definition have had a DATA-TYPE of either 3270-DATA or SCS-DATA
and a RESPONSE-FLAG value of either ERROR-RESPONSE or ALWAYS-
RESPONSE). In this case the possible values of RESPONSE-FLAG
are:

Response-Flag Name	Code	Meaning
POSITIVE-RESPONSE	0x00	The previous message was received and executed successfully with no errors.
NEGATIVE-RESPONSE	0x01	The previous message was received but an error(s) occurred while processing it.

Accompanying status information will be found in the data
portion of the message.

For any other values of the DATA-TYPE field, the RESPONSE-FLAG
field must be ignored by the receiver and should be set to 0x00
by the sender.

8.1.4 SEQ-NUMBER Field

The SEQ-NUMBER field is only used when the RESPONSES function
has been agreed to. It contains a 2 byte binary number, and is
used to correlate positive and negative responses to the data
messages for which they were intended. See the section
entitled "The RESPONSES Function" for further information.
When the RESPONSES function is not agreed to, this field should
always be set to 0x0000 by the sender and ignored by the
receiver.

9. Basic TN3270E

As has been stated earlier, whether or not the use of each of the
TN3270E functions is allowed on a session is negotiated when the
connection is established. It is possible that none of the functions
are agreed to (in this case, the function-list in the FUNCTIONS
REQUEST and FUNCTIONS IS commands is null). This mode of operation
is referred to as "basic TN3270E". Note that, since neither the
SCS-CTL-CODES function nor the DATA-STREAM-CTL function is agreed to,
basic TN3270E refers to terminal sessions only.

Basic TN3270E requires the support of only the following TN3270E
header values:

```
    Header field          Value
    ------------          -----
     DATA-TYPE            3270-DATA
     DATA-TYPE            NVT-DATA
```

The REQUEST-FLAG, RESPONSE-FLAG and SEQ-NUMBER fields are not used in
basic TN3270E.

9.1 3270 Mode and NVT Mode

   At any given time, a TN3270E connection can be considered to be
   operating in either "3270 mode" or "NVT mode". In 3270 mode, each
   party may send data messages with the DATA-TYPE flag set to 3270-
   DATA; sending a DATA-TYPE flag set to NVT-DATA constitutes a
   request to switch modes. In NVT mode, each party may send data
   messages with the DATA-TYPE flag set to NVT-DATA; sending 3270-
   DATA is a request to switch modes. The connection is initially in
   3270 mode when TN3270E operation is successfully negotiated. When
   a party receives a message with a DATA-TYPE different from the
   mode it is operating in, the mode of operation for the connection
   is switched. Switching modes results in the client performing the
   equivalent of a 3270 Erase/Reset operation, as described in [5],
   using the default partition (screen) size. The server cannot
   assume the client preserves any attributes of the previous
   environment across a mode switch.

   Note that even when sending NVT-DATA, each side should buffer data
   until an entire message is built (for the client, this would
   normally mean until the user presses Enter). At that point, a
   complete TN3270E data message should be built to transmit the NVT
   data.

   Typically, NVT data is used by a server to interact with the user
   of a client. It allows the server to do this using a simple NVT
   data stream, instead of requiring a 3270 data stream. An example
   would be a server which displays a list of 3270 applications to
   which it can connect the client. The server would use NVT data to
   display the list and read the user's choice. Then the server
   would connect to the application, and begin the exchange of 3270
   data between the application and the client.

10. Details of Processing TN3270E Functions

   Agreement by both parties to a specific function in the FUNCTIONS
   REQUEST function-list implies agreement by each party to support a
   related set of values in the TN3270E header. It also implies a
   willingness to adhere to the rules governing the processing of data
   messages with regard to the agreed upon function. Either party that
   fails to accept header values associated either with agreed upon
   functions or with basic TN3270E, or attempts to use header values
   associated with a function that is not a part of basic TN3270E and
   was not agreed upon, will be considered non-conforming and in
   violation of the protocol. The following sections detail for each
   TN3270E function the associated header values and processing rules.

   10.1 The SCS-CTL-CODES Function

      This function can only be supported on a 3270 printer session.

      Agreement to support this function requires that the party support
      the following TN3270E header values:

           Header field   Value
           ------------   -----
            DATA-TYPE   SCS-DATA

      A client representing a printer device uses this function to
      indicate its willingness to accept a data stream that includes SCS
      control codes. For the purposes of NVT mode versus 3270 mode,
      SCS-DATA should be treated exactly like 3270-DATA (i.e., it can
      cause a switch from NVT mode to 3270 mode).

      When a printer device-type has been negotiated, either the SCS-
      CTL-CODES function or the DATA-STREAM-CTL function, or both, must
      be negotiated. This enables the server to know when it should and
      should not accept a session with a host application on behalf of
      the client. If only the SCS-CTL-CODES function is agreed to, then
      the server will not establish sessions with host applications that
      would send 3270 data stream control. If both SCS-CTL-CODES and
      DATA-STREAM-CTL are agreed to, then the server will establish
      sessions both with host applications that would send SCS control
      codes and with those that would send 3270 orders.

   10.2 The DATA-STREAM-CTL Function

      This function can only be supported on a 3270 printer session.

      Agreement to support this function requires that the party support
      the following TN3270E header values:

```
        Header field          Value
        ------------          -----
          DATA-TYPE           3270-DATA
```

A client representing a printer device uses this function to
indicate its willingness to accept a data stream that includes
3270 orders and attributes.

When a printer device-type has been negotiated, either the SCS-
CTL-CODES function or the DATA-STREAM-CTL function, or both, must
be negotiated. This enables the server to know when it should and
should not accept a session with a host application on behalf of
the client. If only the DATA-STREAM-CTL function is agreed to,
then the server will not establish sessions with host applications
that would send SCS control codes in a data stream. If both SCS-
CTL-CODES and DATA-STREAM-CTL are agreed to, then the server will
establish sessions both with host applications that would send SCS
control codes and with those that would send 3270 orders.

10.3 The BIND-IMAGE Function

This function can only be supported when the TN3270E server
represents SNA terminals and printers.

Agreement to support this function requires that the party support
the following TN3270E header values:

```
        Header field          Value
        ------------          -----
          DATA-TYPE           BIND-IMAGE
          DATA-TYPE           UNBIND
          DATA-TYPE           SSCP-LU-DATA
```

When BIND-IMAGE is in effect, the server must inform the client
when an SNA session has been established with a host application,
and when such a session has been terminated. It uses DATA-TYPE
values of BIND-IMAGE and UNBIND to convey this information.

When establishing an SNA session on behalf of a client, the server
will receive a Bind RU from the host application. It will also
receive a Start Data Traffic RU. Once both of these have been
responded to positively by the server, it must then inform the
client of the presence of this session by sending it a data
message with the DATA-TYPE flag set to BIND-IMAGE. The data
portion of this message must contain the bind image exactly as it
was received in the Bind RU that the server accepted on behalf of
the client.

When an SNA session between the server and a host application is
terminated, the server should send a data message to the client
with the DATA-TYPE flag set to UNBIND. If the server was notified
of the session termination via an SNA Unbind RU, it should include
the Unbind reason code in the data portion of the message it sends
to the client. If the server itself requested the SNA session
termination (for example, as part of SYSREQ key processing), it
should set the data portion of the UNBIND message to 0x01,
indicating "normal end of session".

Another aspect of the BIND-IMAGE function alters the allowable
DATA-TYPE flag values slightly from the behavior described in the
section entitled "Basic TN3270E". When BIND-IMAGE is in effect,
data messages with DATA-TYPE set to 3270-DATA or SCS-DATA are not
allowed before the first BIND-IMAGE is received by the client;
only SSCP-LU-DATA or NVT-DATA can be used to transmit user-
oriented data. The same applies to data messages exchanged after
an UNBIND is sent and before another BIND-IMAGE is received by the
client. Once the client receives a BIND-IMAGE data message, the
allowable DATA-TYPE values include 3270-DATA and/or SCS-DATA,
depending on whether a terminal or printer device-type was
negotiated, and whether a printer client agreed to DATA-STREAM-CTL
or SCS-CTL-CODES, or both. (See the section entitled "The SYSREQ
Function" for further discussion of the SSCP-LU session in an SNA
environment.)

10.4 The RESPONSES Function

This function can be supported for both terminal and printer
sessions connected to both SNA and non-SNA servers.

Agreement to support this function requires that the party support
the following TN3270E header values:

```
    Header field         Value
    ------------         -----
     DATA-TYPE           RESPONSE
     DATA-TYPE           REQUEST
     RESPONSE-FLAG       -all values-
     REQUEST-FLAG        ERR-COND-CLEARED
     SEQ-NUMBER          binary values from 0-32767
```

Whenever a data message is sent with a DATA-TYPE of either SCS-
DATA or 3270-DATA, the sender must set the RESPONSE-FLAG field to
either NO-RESPONSE, ERROR-RESPONSE, or ALWAYS-RESPONSE. It is
anticipated that the client side will normally set RESPONSE-FLAG
to NO-RESPONSE. The server, if it represents an SNA device,
should set RESPONSE-FLAG to reflect the response value set in the

RH of the RU that generated this data message - Definite Response
resulting in a RESPONSE-FLAG value of ALWAYS-RESPONSE, Exception
Response resulting in ERROR-RESPONSE being set, and No Response
causing a setting of NO-RESPONSE. A non-SNA server should set
RESPONSE-FLAG to ERROR-RESPONSE.

In addition, the sender must keep a count of the messages with a
DATA-TYPE of 3270-DATA or SCS-DATA that it sends on a given
session. This counter should start at zero for the first such
message, and be incremented by one for each subsequent message.
If the counter reaches the maximum of 32767, it should be
restarted at zero. The sender should place this value in the
SEQ-NUMBER field of the TN3270E header before it sends the
message. Note that the SEQ-NUMBER field must be set regardless of
the value of the RESPONSE-FLAG field.

10.4.1 Response Messages

   Whenever a data message with a DATA-TYPE of either SCS-DATA or
   3270-DATA is received, the receiver must attempt to process the
   data in the data portion of the message, then determine whether
   or not it should send a data message with a DATA-TYPE of
   RESPONSE. If the data message it has just processed had a
   RESPONSE-FLAG value of NO-RESPONSE, or if it had a value of
   ERROR-RESPONSE and there were no errors encountered while
   processing the data, then no RESPONSE type message should be
   sent. Otherwise, a data message should be sent in which the
   header DATA-TYPE field is set to RESPONSE, and in which the
   SEQ-NUMBER field is a copy of the SEQ-NUMBER field from the
   message to which this response corresponds. The RESPONSE-FLAG
   field in this header must have a value of either POSITIVE-
   RESPONSE or NEGATIVE-RESPONSE. A POSITIVE-RESPONSE should be
   sent if the previously processed message's header specified
   ALWAYS-RESPONSE and no errors were encountered in processing
   the data. A NEGATIVE-RESPONSE should be sent when

   1) the previously processed message specified ERROR-RESPONSE
      or ALWAYS-RESPONSE and

   2) some kind of error occurred while processing the data.

   Normally only the client will be constructing and sending these
   RESPONSE messages. A negative response sent by the client to
   the server is the equivalent of a Unit Check Status [7]. All
   references to device status and sense codes in this section
   rely on [7].

The data portion of a RESPONSE message must consist of one byte
of binary data. The value of this byte gives a more detailed
account of the results of having processed the previously
received data message. The possible values for this byte are:

For a RESPONSE-FLAG value of POSITIVE-RESPONSE -

Value	Meaning
0x00	Successful completion (when sent by the client, this is equivalent to "Device End").

For a RESPONSE-FLAG value of NEGATIVE-RESPONSE -

Value	Meaning
0x00	An invalid 3270 command was received (equivalent to "Command Reject").
0x01	Printer is not ready (equivalent to "Intervention Required").
0x02	An illegal 3270 buffer address or order sequence was received (equivalent to "Operation Check").
0x03	Printer is powered off or not connected (equivalent to "Component Disconnected").

When the server receives any of the above responses, it should
pass along the appropriate information to the host application.
The appropriate information is determined by whether the server
represents an SNA or a non-SNA device.

An SNA server should pass along a POSITIVE-RESPONSE from the
client as an SNA positive Response Unit to the host
application. It should translate a NEGATIVE-RESPONSE from the
client into an SNA negative Response Unit in which the Sense
Data Indicator bit is on and which contains one of the
following sense codes:

RESPONSE-FLAG	Equivalent	SNA Sense Code
0x00	Command Reject	0x10030000
0x01	Intervention Required	0x08020000
0x02	Operation Check	0x10050000
0x03	Component Disconnected	0x08310000

A non-SNA server should pass along a POSITIVE-RESPONSE from the
client by setting the Device End Status bit on. It should
reflect a NEGATIVE-RESPONSE from the client by setting the Unit
Check Status Bit on, and setting either the Command Reject,
Intervention Required, or Operation Check Sense bit on when
responding to the Sense command.

In the case of Intervention Required or Component Disconnected
being passed by the server to the host application, the host
would normally refrain from sending any further data to the
printer. If and when the error condition at the client has
been resolved, the client must send to the server a data
message whose header DATA-TYPE field is set to REQUEST, and
whose REQUEST-FLAG is set to ERR-COND-CLEARED. Note that this
message has no data portion. Upon receipt of this message, the
server should pass along the appropriate information to the
host application so that it may resume sending printer output.
Again, the form of this information depends on whether the
server represents an SNA or a non-SNA device.

An SNA server should reflect an ERR-COND-CLEARED to the host
application by sending an SNA LUSTAT RU with one of the
following sense codes:

  - if the previous error condition was an Intervention
    Required, the server should send sense code 0x00010000

  - if the previous error condition was Component
    Disconnected, the server should send sense code 0x082B0000

A non-SNA server should set the corresponding bits in the
Ending Status and Sense Condition bytes.

   10.5 The SYSREQ Function

      This function can only be supported when the TN3270E server
      represents SNA devices.

      Agreement to support this function requires that the party support
      the following TN3270E header values:

         Header field   Value
         ------------   -----
          DATA-TYPE   SSCP-LU-DATA

      The 3270 SYSREQ key can be useful in an SNA environment when the
      ATTN key is not sufficient to terminate a process. (See the
      section entitled "The 3270 ATTN Key" for more information.)

   10.5.1 Background

      In SNA, there is a session between the host application (the
      PLU, or Primary Logical Unit) and the TN3270E server
      representing the client (the SLU, or Secondary Logical Unit).
      This is referred to as the PLU-SLU session, and it is the one
      on which normal communications flow. There is also a session
      between the host telecommunications access method (the SSCP, or
      System Services Control Point) and the SLU, and it is referred
      to as the SSCP-LU session. This session is used to carry
      various control information and is normally transparent to the
      user; normal 3270 data stream orders are not allowed in this
      data. For more information, refer to [7].

      The terminal display and keyboard are usually "owned" by the
      PLU-SLU session, meaning any data the user types is sent to the
      host application. The SYSREQ key is used to toggle ownership
      of the keyboard and display between the PLU-SLU session and the
      SSCP-LU session. In other words, the user is able to press
      SYSREQ and then communicate directly with the host SSCP. The
      user may then enter any valid Unformatted Systems Services
      commands, which are defined in the USS table associated with
      the SLU. The most common USS command users employ is "LOGOFF,"
      which requests that the SSCP immediately terminate the PLU-SLU
      session. The usual reason for requesting such an action is
      that the host application (the PLU) has stopped responding
      altogether.

      Whenever the keyboard and display are owned by the SSCP-LU
      session, no data is allowed to flow in either direction on the
      PLU-SLU session. Once "in" the SSCP-LU session, the user may
      decide to switch back to the PLU-SLU session by again pressing

the SYSREQ key.

   10.5.2 TN3270E Implementation of SYSREQ

      The design of some TN3270E servers allows them to fully support
      the SYSREQ key because they are allowed to send USS commands on
      the SSCP-LU session. Other TN3270E servers operate in an
      environment which does not allow them to send USS commands to
      the SSCP; this makes full support of the SYSREQ key impossible.
      For such servers, TN3270E provides for emulation of a minimal
      subset of functions, namely, for the sequence of pressing
      SYSREQ and typing LOGOFF that many users employ to immediately
      terminate the PLU-SLU session.

      The Telnet Abort Output (AO) command is the mechanism used to
      implement SYSREQ key support in TN3270E because, in a real SNA
      session, once the user presses the SYSREQ key, the host
      application is prevented from sending any more output to the
      terminal (unless the user presses SYSREQ a second time), but
      the user's process continues to execute.

      In order to implement SYSREQ key support, TN3270E clients that
      have agreed to the SYSREQ function should provide a key (or
      combination of keys) that is identified as mapping to the 3270
      SYSREQ key. When the user presses this key(s), the client
      should transmit a Telnet AO command to the server.

      Upon receipt of the AO command, a TN3270E server that has
      agreed to the SYSREQ function should enter what will be loosely
      termed "suspended mode" for the connection. If a server that
      has not agreed to the SYSREQ function receives an AO command,
      it should simply ignore it. Any attempt by the host
      application to send data to the client while the connection is
      "suspended" should be responded to by the server with a
      negative response, sense code 0x082D, indicating an "LU Busy"
      condition. The server should not transmit anything to the
      client on behalf of the host application. While the connection
      is "suspended," any data messages (except TN3270E responses)
      exchanged between the client and server should have the DATA-
      TYPE flag set to SSCP-LU-DATA.

      At this point, the behavior of the server depends upon whether
      or not it is allowed to send USS commands on the SSCP-LU
      session. Servers that have this ability should simply act as a
      vehicle for passing USS commands and responses between the
      client and the SSCP.

Servers that are not allowed to send USS commands on the SSCP-
LU session should behave as follows:

- if the user transmits the string LOGOFF (upper or lower case),
  the server should send an Unbind SNA RU to the host
  application. This will result in termination of the PLU-SLU
  session. If the BIND-IMAGE function was agreed upon, then
  the server should also send a data message to the client with
  the DATA-TYPE flag set to UNBIND and the data portion set to
  0x01.

- if the user transmits anything other than LOGOFF, the server
  should respond with the string "COMMAND UNRECOGNIZED" to the
  client. The server should not send anything to the host
  application on behalf of the client.

  Regardless of which kind of server is present (i.e., whether or
  not it may send USS commands on the SSCP-LU session), while the
  connection is suspended, the user may press the "SYSREQ" key
  again. This will result in the transmission of another AO to
  the server. The server should then send to the host
  application an LUSTAT RU with a value of 0x082B indicating
  "presentation space integrity lost". The server will then
  "un-suspend" the Telnet connection to the client, meaning it
  will allow the host application to once again send data to the
  client.

11. The 3270 ATTN Key

The 3270 ATTN key is interpreted by many host applications in an SNA
environment as an indication that the user wishes to interrupt the
execution of the current process. The Telnet Interrupt Process (IP)
command was defined expressly for such a purpose, so it is used to
implement support for the 3270 ATTN key. This requires two things:

- TN3270E clients should provide as part of their keyboard
  mapping a single key or a combination of keys that map to
  the 3270 ATTN key. When the user presses this key(s), the
  client should transmit a Telnet IP command to the server.

- TN3270E servers should translate the IP command received from
  a TN3270E client into the appropriate form and pass it along
  to the host application as an ATTN key. In other words, the
  server representing an SLU in an SNA session should send
  a SIGNAL RU to the host application.

The ATTN key is not supported in a non-SNA environment; therefore, a
TN3270E server representing non-SNA 3270 devices should ignore any
Telnet IP commands it receives from a client.

12. 3270 Structured Fields

3270 structured fields provide a much wider range of features than
"old-style" 3270 data, such as support for graphics, partitions and
IPDS printer data streams. It would be unreasonable to expect all
TN3270E clients to support all possible structured field functions,
yet there must be a mechanism by which those clients that are capable
of supporting some or all structured field functions can indicate
their wishes.

The design of 3270 structured fields provides a convenient means to
convey the level of support (including no support) for the various
structured field functions. This mechanism is the Read Partition
Query command, which is sent from the host application to the device.
The device responds with a Query Reply structured field(s) listing
which, if any, structured field functions it supports.

The Query Reply is also used to indicate some device capabilities
which do not require the use of structured fields, such as extended
color support and extended highlighting capability. Most host
applications will use Read Partition Query to precisely determine a
device's capabilities when there has been some indication that the
device supports the "extended data stream".

Therefore, all TN3270E clients that negotiate a terminal device-type
that contains a "-E" suffix, the DYNAMIC terminal type, or a printer
device-type, must be able to respond to a Read Partition Query
command. Note that these clients must support both the Read
Partition Query (Type 02), and all forms of the Read Partition Query
List (Type 03).

13. Implementation Guidelines

13.1 3270 Data Stream Notes

Implementors of TN3270E clients should note that the command codes
for the various 3270 Read and Write commands have different values
depending on how the server is connected to the host (local versus
remote, SNA versus non-SNA). Clients should be coded to check for
the various possible values if they wish to be compatible with the
widest range of servers. See [7] for further details.

13.2 Negotiation of the TN3270E Telnet Option

   Since TN3270E is a Telnet Option governed by [8], both client and
   server are free to attempt to initiate negotiation of TN3270E by
   sending a DO TN3270E command. However, just as is usually the
   case with the Telnet DO TERMINAL-TYPE, it is anticipated that the
   server will normally be the one sending the DO TN3270E, and the
   client will be responding with a WILL or a WON'T TN3270E.

13.3 A "Keep-alive" Mechanism

   In many environments, it is very helpful to have in place a
   mechanism that allows timely notification of the loss of a 3270
   session. TN3270E does not require that any form of keep-alive
   mechanism be employed by either clients or servers, but
   implementors wishing to support such a mechanism should consider
   the following guidelines.

   There are at least two possible means of providing a keep-alive
   mechanism in TN3270E: the Telnet IAC NOP command [8], and the
   Telnet DO TIMING-MARK option [9]. Both methods have their
   advantages and disadvantages. It is recommended that TN3270E
   clients and servers that support keep-alives should accept both
   NOPs and TIMING-MARKs, and that both sides should always respond
   to TIMING-MARKs.

   Note that both clients and servers could be configured to
   "actively" implement keep-alives. That is, both sides could send
   a TIMING-MARK or a NOP in order to determine whether or not the
   partner is still alive. Alternatively, network administrators may
   wish to configure only one side to send TIMING-MARKs or NOPs; in
   this case, the other side would be a "passive" participant which
   simply responds to the keep-alives it receives.

   Implementors who want their code to be capable of being an
   "active" keep-alive participant should make their client or server
   configurable so that administrators can set which, if any, keep-
   alive mechanism should be employed, and how often the NOP or
   TIMING-MARK should be sent on each session.

   Upon failure of a session on which keep-alives are used, both
   parties should make the proper notifications. A client should
   give the user some indication of the failure, such as an error
   code in the Operator Information Area of the screen. A server
   should notify the host application that the session has been
   terminated, for example by sending an UNBIND with type CLEANUP in
   an SNA environment.

   13.4 Examples

      The following example shows a TN3270E-capable server and a
      traditional tn3270 client establishing a connection:

         Server: IAC DO TN3270E
         Client: IAC WON'T TN3270E
         Server: IAC DO TERMINAL-TYPE
         Client: IAC WILL TERMINAL-TYPE
         Server: IAC SB TERMINAL-TYPE SEND IAC SE
         Client: IAC SB TERMINAL-TYPE IS IBM-3278-2 IAC SE
         Server: IAC DO EOR IAC WILL EOR
         Client: IAC WILL EOR IAC DO EOR
         Server: IAC DO BINARY IAC WILL BINARY
         Client: IAC WILL BINARY IAC DO BINARY
            (3270 data stream is exchanged)

      The following example shows a TN3270E-capable server and a
      TN3270E-capable client establishing a generic pool (non-specific)
      terminal session:

         Server: IAC DO TN3270E
         Client: IAC WILL TN3270E
         Server: IAC SB TN3270E SEND DEVICE-TYPE IAC SE
         Client: IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3278-2 IAC SE
         Server: IAC SB TN3270E DEVICE-TYPE IS IBM-3278-2 CONNECT
                        anyterm IAC SE
         Client: IAC SB TN3270E FUNCTIONS REQUEST RESPONSES IAC SE
         Server: IAC SB TN3270E FUNCTIONS IS RESPONSES IAC SE
            (3270 data stream is exchanged)

      The following example shows a TN3270E-capable server and a
      TN3270E-capable client establishing a terminal session where the
      client requests a specific device-name:

         Server: IAC DO TN3270E
         Client: IAC WILL TN3270E
         Server: IAC SB TN3270E SEND DEVICE-TYPE IAC SE
         Client: IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3278-5-E
                        CONNECT myterm IAC SE
         Server: IAC SB TN3270E DEVICE-TYPE IS IBM-3278-5-E CONNECT
                        myterm IAC SE
         Client: IAC SB TN3270E FUNCTIONS REQUEST RESPONSES
                        BIND-IMAGE IAC SE
         Server: IAC SB TN3270E FUNCTIONS IS RESPONSES BIND-IMAGE
                        IAC SE
            (3270 data stream is exchanged)

The following example shows a TN3270E-capable server and a
TN3270E-capable client attempting to establish a terminal session;
multiple attempts are necessary because the device-name initially
requested by the client is already in use:

```
Server:  IAC DO TN3270E
Client:  IAC WILL TN3270E
Server:  IAC SB TN3270E SEND DEVICE-TYPE IAC SE
Client:  IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3278-5
                 CONNECT myterm IAC SE
Server:  IAC SB TN3270E DEVICE-TYPE REJECT REASON
                 DEVICE-IN-USE IAC SE
Client:  IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3278-2
                 CONNECT herterm IAC SE
Server:  IAC SB TN3270E DEVICE-TYPE IS IBM-3278-2 CONNECT
                 herterm IAC SE
Client:  IAC SB TN3270E FUNCTIONS REQUEST RESPONSES IAC SE
Server:  IAC SB TN3270E FUNCTIONS IS RESPONSES IAC SE
   (3270 data stream is exchanged)
```

The following example shows a TN3270E-capable server and a
TN3270E-capable client establishing a printer session where the
client requests a specific device-name, and where some amount of
3270 function negotiation is required before an agreement is
reached:

```
Server:  IAC DO TN3270E
Client:  IAC WILL TN3270E
Server:  IAC SB TN3270E SEND DEVICE-TYPE IAC SE
Client:  IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3287-1 CONNECT
                 myprt IAC SE
Server:  IAC SB TN3270E DEVICE-TYPE IS IBM-3287-1 CONNECT
                 myprt IAC SE
Client:  IAC SB TN3270E FUNCTIONS REQUEST DATA-STREAM-CTL IAC
Server:  IAC SB TN3270E FUNCTIONS REQUEST DATA-STREAM-CTL
                 RESPONSES IAC SE
Client:  IAC SB TN3270E FUNCTIONS REQUEST DATA-STREAM-CTL IAC
Server:  IAC SB TN3270E FUNCTIONS IS DATA-STREAM-CTL IAC SE
   (3270 data stream is exchanged)
```

The following example shows a TN3270E-capable server and a
TN3270E-capable client establishing first a generic terminal
session, then a printer session where the "partner" printer for
the assigned terminal is requested:

```
Server:  IAC DO TN3270E
Client:  IAC WILL TN3270E
Server:  IAC SB TN3270E SEND DEVICE-TYPE IAC SE
```

```
       Client:  IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3278-2 IAC SE
       Server:  IAC SB TN3270E DEVICE-TYPE IS IBM-3278-2 CONNECT
                        termXYZ IAC SE
       Client:  IAC SB TN3270E FUNCTIONS REQUEST RESPONSES IAC SE
       Server:  IAC SB TN3270E FUNCTIONS IS RESPONSES IAC SE
          (3270 data stream is exchanged)
                .            .
                .            .
                .            .
          (user decides to request a printer session,
           so client again connects to Telnet port on server)
       Server:  IAC DO TN3270E
       Client:  IAC WILL TN3270E
       Server:  IAC SB TN3270E SEND DEVICE-TYPE IAC SE
       Client:  IAC SB TN3270E DEVICE-TYPE REQUEST IBM-3287-1
                        ASSOCIATE termXYZ IAC SE
       Server:  IAC SB TN3270E DEVICE-TYPE IS IBM-3287-1 CONNECT
                        termXYZ's-prt IAC SE
       Client:  IAC SB TN3270E FUNCTIONS REQUEST SCS-CTL-CODES
                        RESPONSES IAC SE
       Server:  IAC SB TN3270E FUNCTIONS IS SCS-CTL-CODES RESPONSES
                        IAC SE
          (3270 data stream is exchanged)
```

14. Security Considerations

Security issues are not addressed in this document. It is
anticipated that once authentication mechanisms have become well
established, use of them can be made by TN3270E. One of the
important uses of authentication would be to answer the question of
whether or not a given user should be allowed to "use" a specific
terminal or printer device-name.

15. References

   [1] Rekhter, J., "Telnet 3270 Regime Option", RFC 1041, IBM
       Corporation, January 1988.

   [2] VanBokkelen, J., "Telnet Terminal-Type Option", RFC 1091, FTP
       Software, Inc., February 1989.

   [3] Postel, J., and J. Reynolds, "Telnet Binary Transmission", STD
       27, RFC 856, USC/Information Sciences Institute, May 1983.

   [4] Postel, J., "Telnet End of Record Option", RFC 885, USC/
       Information Sciences Institute, December 1983.

   [5] "3270 Information Display System - Data Stream Programmer's
       Reference", publication number GA24-0059, IBM Corporation.

   [6] "SNA Formats", publication number GA27-3136, IBM Corporation.

   [7] "3174 Establishment Controller Functional Description",
       publication number GA23-0218, IBM Corporation.

   [8] Postel, J., and J. Reynolds, "Telnet Protocol Specification", STD
       8, RFC 854, USC/Information Sciences Institute, May 1983.

   [9] Postel, J., and J. Reynolds, "Telnet Timing Mark Option", STD 31,
       RFC 860, USC/Information Sciences Institute, May 1983.

16. Author's Note

   Portions of this document were drawn from the following sources:

    - A White Paper written by Owen Reddecliffe, WRQ Corporation,
      October 1991.

    - Experimental work on the part of Cleve Graves and Michelle
      Angel, OpenConnect Systems, 1992 - 1993.

    - Discussions at the 1993 IETF meetings.

    - Discussions on the "TN3270E" list, 1993-94.

17. Author's Address

   Bill Kelly
   Division of University Computing
   144 Parker Hall
   Auburn University, AL 36849

   Phone: (205) 844-4512
   EMail: kellywh@mail.auburn.edu