

## TN3270 Current Practices

### Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Abstract

This document describes the existing implementation of transferring 3270 display terminal data using currently available telnet capabilities. The name traditionally associated with this implementation is TN3270.

Information is provided to aid in the implementation of TN3270 servers as well as client terminal emulators.

The following areas pertaining to TN3270 implementations are covered in this document:

1. the telnet options negotiated to transition from a NVT ASCII state to a TN3270 state ready to process incoming 3270 data stream commands
2. the method for sending and receiving 3270 data
3. the method of handling some special keys known as SYSREQ and ATTN using current available telnet commands
4. the events that will transition a TN3270 session back to an NVT session

### Table of Contents

1. Motivation . . . . .	2
2. Background . . . . .	2
3. Telnet Options and Commands Used . . . . .	4
4. Connection Negotiation . . . . .	4
4.1 3270 Regime Option . . . . .	6
4.2 Suppress Go Ahead Option . . . . .	6
4.3 Echo Option . . . . .	6
4.4 Timing Mark Option . . . . .	7

- 5. Testing for session presence . . . . . 7
- 6. Handling 3270 data . . . . . 7
- 7. 3270 Structured Fields . . . . . 8
- 8. The 3270 ATTN (Attention) Key . . . . . 8
- 9. The 3270 SYSREQ Key . . . . . 9
- 10. Items not addressed by TN3270 . . . . . 10
- 11. References . . . . . 10
- 12. Security Considerations . . . . . 11
- 13. Author's Note . . . . . 11
- 14. Author's Address . . . . . 12

1. Motivation

3270 display terminal data differs from traditional display terminal data in that it is block mode and uses EBCDIC instead of ASCII character representation. These two differences are the primary reason for the differentiation of TN3270 from standard Telnet in this document.

2. Background

Existing complex IBM 3270 display terminal networks are not easily integrated with the increasing number of multi-platform networking environments, specifically TCP/IP. These complex networks include terminals attached to a 3270 host using SNA (Systems Network Architecture) and non-SNA connections. To address the issue of easily connecting display terminals to 3270 hosts using IP networks, several vendors have introduced telnet servers that provide TCP/IP users a connection to existing IBM mainframes by supporting display terminal emulation using a subset of the existing telnet protocol. Telnet servers may exist on the host itself, or be connected to the host using SNA or non-SNA methods.

IBM terminals are generically referred to as 3270's which includes a broad range of terminals and devices, not all of which actually begin with the numbers 327x.

3270 terminals in the IBM SNA network environment have two sessions with the host computer application. One is used for communicating with the host application, the other is used for communicating with the SSCP (System Services Control Point) that links the terminal with the appropriate host computer. For the purposes of TN3270, this distinction is not apparent or relevant since there is actually only a single telnet session with the host computer or server. On an IBM SNA network, the 3270 terminal has a special key that toggles between the two sessions (SYSREQ). A brief discussion on how some telnet servers deal with this is included.

In an SNA environment, a client session is identified by a Logical Unit (LU) name. In a non-SNA environment, there is not a LU name associated with a client session. The closest thing to a LU name in the TN3270 environment is the client's IP address. Although some telnet servers are connected to the host using SNA, TN3270 clients using these servers have no defined way to determine the LU name associated with the session.

Telnet servers that exist in non-SNA environments do not have to be concerned about providing TN3270 clients with support for the SNA functions described in this document.

TN3270 does not support typical SNA responses and is classified as a non-SNA protocol. A TN3270 emulator is not aware or concerned about how the telnet server is connected to a 3270 host application.

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.

Some typical "SNA" functions such as the SYSREQ and ATTN keys have been mapped to existing telnet commands and are supported by some telnet server implementations.

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]. This negotiation and the resulting data flow will be described below.

RFC 1041 [1] attempted to standardize the method of negotiating 3270 terminal support by defining the 3270 Regime Telnet Option. Historically, very few developers and vendors ever implemented RFC 1041.

All references in this document to the 3270 datastream, SNA versus non-SNA operation, 3270 datastream commands, orders, structured fields and the like rely on [6].

References to SNA Request and Response Units rely on [7]. References to SNA and SSCP rely on [12].

### 3. Telnet Options and Commands Used

TN3270 makes use of existing Telnet options and does not define any additional options or commands.

Telnet option	Value (decimal)
-----	-----
BINARY	0
TERMINAL-TYPE	24
EOR	25

Additional options may be used during a TN3270 session and are interpreted as per their respective RFCs. These are [1] 3270-REGIME, [8] SUPPRESS-GO-AHEAD, [9] ECHO and [10] TIMING-MARK. Other options should be rejected unless they are specifically handled by the client for NVT mode.

Commands that may be encountered during a TN3270 session and are described in RFC 854 [11] include NOP, BREAK and Interrupt Process.

### 4. Connection Negotiation

The following example shows a TN3270-capable server and a TN3270 client establishing a connection:

The TCP/IP port used to connect with is 23 (Telnet).

At any place before and during the TN3270 connection negotiation process, other telnet commands and data may be transferred and will be interpreted under the existing telnet state. Some existing TN3270 servers start a client connection using an NVT telnet dialog to establish parameters needed to complete the TN3270 connection to the desired host.

The order of negotiating terminal type, EOR and BINARY is not significant, this example shows a typical TN3270 connection.

Server: IAC DO TERMINAL-TYPE

Client: IAC WILL TERMINAL-TYPE

Server: IAC SB TERMINAL-TYPE SEND IAC SE

Client: IAC SB TERMINAL-TYPE IS <terminal type>IAC SE

where <terminal type> is a string consisting of terminal model, type and support of enhanced attribute bytes; an example is IBM-3278-2. The acceptable values are listed in RFC 1340, Assigned

Numbers [5]. Other values are in use that do not exist in [5].

The -2 following 3278 designates the alternate screen size. 3270 terminals have the ability to switch between the standard (24x80) screen size and an alternate screen size. Model -2 is 24x80 which is the same as the standard size. Model -3 is 32x80, model -4 is 43x80 and model -5 is 27x132.

Appending the two character string "-E" to the end of the terminal type signifies that the terminal is capable of handling 3270 extended data stream. This is interpreted to mean that the terminal is able to handle structured fields, which are described below. Some telnet server implementations also interpret this to mean that the terminal is capable of handling extended attributes (highlighting, field validation, character set, outlining, etc.) [6].

The 3279 series of terminals is capable of extended attributes while the 3278 series is not.

```
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
Server:  <3270 data stream> IAC EOR
Client:  <3270 data stream> IAC EOR
      .
      .
      .
```

To terminate the connection the socket is closed by one of the session partners. Typically, when the user logs off of the host, the telnet server closes the connection.

If the telnet server wishes to go back to NVT mode, it may issue the following telnet options:

```
Server:  IAC WONT BINARY
Client:  IAC DONT BINARY
```

or

```
Server:  IAC WONT EOR
Client:  IAC DONT EOR
```

Either one of the above two cases causes the connection to not satisfy the requirements for a valid TN3270 session. The telnet client would then process data from the server as though it were NVT ASCII data.

The following examples show how a TN3270 client handles the 3270-REGIME, SUPPRESS-GO-AHEAD, ECHO and TM options.

#### 4.1 3270 Regime Option

Very few servers support the 3270 Regime Telnet Option. If the client does not support this option and responds negatively as shown in the following example, the server will proceed on to the more typical example shown above.

```
Server: IAC DO 3270-REGIME
Client: IAC WONT 3270-REGIME
Normal negotiation:
Server: IAC DO TERMINAL-TYPE
... (see above)
```

#### 4.2 Suppress Go Ahead Option

The Suppress Go Ahead option [8] is requested by some servers. The Suppress Go Ahead option RFC lists the default as being go aheads are transmitted to signal the receiver to begin transmitting. Since TN3270 negotiates binary and end-of-record and is a block mode protocol, the telnet go ahead character is not sent. Most servers do not negotiate this option even though they do not use the telnet go ahead character.

```
Server: IAC DO SUPPRESS-GO-AHEAD
Client: IAC WILL SUPPRESS-GO-AHEAD
```

#### 4.3 Echo Option

The Echo option [9] is negotiated by those servers that make use of the telnet NVT mode to allow the user to enter information prior to negotiating the options necessary for TN3270. This information includes but is not limited to user identification, password and destination 3270 host. Some servers accept the default for this option which is for the client to not do a local echo of characters the user enters at the keyboard. This allows the server to decide if it should echo characters back to the client (or not in the case of password). Echoing characters back to the client causes slow response time since every character is typically echoed individually. Because of this, some servers negotiate for the client to do it's own local echoing (except for passwords). The following example illustrates this case.

```
Server: IAC DO ECHO
Client: IAC WILL ECHO
      (Client does local display of all characters)
Server: IAC WONT ECHO
Client: IAC DONT ECHO
      (Client enters password - not locally displayed or remotely
echoed)
Server: IAC DO ECHO
Client: IAC WILL ECHO
      (Client resumes local display of all characters)
```

#### 4.4 Timing Mark Option

The Timing Mark option [10] is used by some servers to test for the continued presence of a TN3270 client. The following example will assure the server the client is still alive.

```
Server: IAC DO TIMING-MARK
Client: IAC WONT TIMING-MARK
```

#### 5. Testing for session presence

The NOP command (hexadecimal F1) [11] is used by some servers to test for the continued presence of a TN3270 client. If a client has terminated abnormally, TCP/IP send errors will occur. The Timing Mark option, described above, is also used to test for presence.

```
Server: IAC NOP
Client: <ignore / no response>
```

#### 6. Handling 3270 data

The 3270 data stream consists of a command and its associated data. Commands include but are not limited to erase screen, erase and write to screen and read current screen; see [6] for a complete description of 3270 commands and parameters.

The reason for negotiating the EOR telnet option [4] is to provide a method for separating these commands since no length information is specified. 3270 commands are interpreted by the telnet client in their entirety. Each 3270 command and possible data is terminated with the IAC EOR sequence.

The Binary option [3] is also required since 3270 data may contain the FF (hexadecimal) or IAC character. When this character is encountered during a TN3270 connection it is handled as per the Binary RFC [3].

## 7. 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 datastreams. A structured field is a 3270 data type that allows non 3270 data to be embedded within 3270 data. Briefly, a structured field consists of the structured field command followed by one or more data blocks. Each data block has a length and a structured field identifier, followed optionally by additional data.

Not every TN3270 client can be expected to support all structured field functions. There must be a mechanism by which those clients that are capable of supporting some or all structured field functions can indicate their wishes. This is typically done by adding "-E" to the end of the terminal type string. That is, when the terminal identifies itself as being able to handle extended attributes, it also is capable of being able to send and receive structured fields.

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 client. The client responds with a Query Reply, listing which, if any, structured field functions it supports.

A TN3270 client that supports structured fields will respond to a Read Partition Query command with the appropriate reply. The sequence of events when a client receives a Read Partition Query and does not support structured fields is left up to the client implementation. Typically clients can identify at least this structured field and reply with a null set.

## 8. The 3270 ATTN (Attention) 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. A majority of the telnet servers currently accept the telnet IAC BREAK (code 243) [11] sequence to signal this event.

Use of this key requires two things:

- The TN3270 clients 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 transmits a Telnet BREAK command to the server.



- The TN3270 servers translate the BREAK command received from a TN3270 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 would send a SIGNAL RU to the host application.

The ATTN key is not supported in a non-SNA environment; therefore, a TN3270 server representing non-SNA 3270 devices ignores any Telnet BREAK commands it receives from a client.

#### 9. The 3270 SYSREQ Key

The 3270 SYSREQ key is useful in an environment where the telnet server is attached to the host using SNA. The SYSREQ key is useful in this environment when the host application becomes locked or the user wishes to terminate the session without closing the Telnet connection.

The Telnet Interrupt Process (IP) command [11] is interpreted by some telnet servers as a SYSREQ key. Other servers recognize the 3270 Test Request key as a SYSREQ key. In an SNA environment, pressing this key toggles the terminal between the host application session and the SSCP session. Usually the user will enter LOGOFF once this key has been pressed to terminate the application session and then select a new host to connect to. Sometimes, if SYSREQ is pressed again, the host application will become unlocked and normal activities may then proceed.

It is entirely up to the telnet server to interpret this command and send the appropriate commands to the host as well as format the resulting host data for display on the telnet client. The data format during the SSCP session is in a slightly different format than normal 3270 data. Since the telnet server has no way to pass this data directly to the telnet client, it must either handle it entirely and ignore SYSREQ events or convert it to 3270 data to present to the client.

To implement SYSREQ key support, TN3270 clients 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 would either transmit a Telnet IP command or Test Request key to the server, depending on the server implementation.

TN3270 servers representing non-SNA 3270 terminals may ignore any Telnet IP commands or Test Request keys they receive from a client.

## 10. Items not addressed by TN3270

There are several items that are not supported by current TN3270 implementations; among them are the following:

- TN3270 provides no capability for 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.
- The 3270 ATTN and SYSREQ keys are not universally supported.
- There is no support for the SNA positive/negative response process. All data that is sent is assumed to either be handled or ignored. The lack of SNA response processing in TN3270 is part of what makes TN3270 efficient.  
A negative response indicates some sort of error at the client while processing the previously received data; this could be caused by the host application building a 3270 datastream that contains an invalid command, or by a mechanical error at the client side, among other things.  
Positive responses indicate processing of the previously received data has completed.
- There is no mechanism by which the client can access the SNA BIND information. The BIND image in a SNA environment contains a detailed description of the session between the telnet server and the host application.
- The connection negotiation does not make it clear whether clients should support 3270 structured fields.

## 11. References

- [1] Rekhter, Y., "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] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.
- [6] "3270 Information Display System - Data Stream Programmer's Reference", publication number GA23-0059, IBM Corporation.
- [7] "Systems Network Architecture - Formats", publication number GA27-3136, IBM Corporation.
- [8] Postel, J., and J. Reynolds, "Telnet Suppress Go Ahead Option", STD 29, RFC 858, USC/Information Sciences Institute, May 1983.
- [9] Postel, J., and J. Reynolds, "Telnet Echo Option", STD 28, RFC 857, USC/Information Sciences Institute, May 1983.
- [10] Postel, J., and J. Reynolds, "Telnet Timing Mark Option", STD 31, RFC 860, USC/Information Sciences Institute, May 1983.
- [11] Postel, J., and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, USC/Information Sciences Institute, May 1983.
- [12] "Systems Network Architecture - Concepts and Products", publication number GC30-3072, IBM Corporation.

## 12. Security Considerations

Security issues are not discussed in this memo.

## 13. 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 March 1993 IETF meeting and TN3270 BOF at Interop August 1993.
- Discussions on the "TN3270E" list, 1993.

14. Author's Address

Jon Penner  
DCA, Inc.  
2800 Oakmont Drive  
Austin, TX 78664

Phone: (512) 388-7090 FAX

E-Mail: [jjp@bscs.com](mailto:jjp@bscs.com)

or [dca/g=Jon/s=Penner/ou=DCAAUS@mhs.attmail.com](mailto:dca/g=Jon/s=Penner/ou=DCAAUS@mhs.attmail.com)