

## 5250 Telnet Interface

### Status of this Memo

This RFC is being distributed in order to document the interface to the IBM 5250 Telnet implementation. This information is being provided for hosts on the Internet that want to support the 5250 work station data stream within the Telnet protocol. This memo provides information for the Internet community. It does not specify any standard. Distribution of this memo is unlimited.

### 1. Introduction

This RFC describes the interface to the IBM 5250 Telnet implementation. The purpose of this memo is to describe the details of the interface so that a person wanting to implement a client Telnet which emulates an IBM 5250 work station would be able to do so. This memo does not describe all of the 5250 commands, aid codes, and other information specific to the 5250 data stream. That information is contained in the IBM 5250 Information Display System, Functions Reference Manual, IBM publication number SA21-9247. Corrections and additions to this manual are documented in this RFC in section 5.

### 2. Telnet Options

No new Telnet options are defined for 5250 mode of operation. However, to enable 5250 mode, both the client and server must agree to at least support the Binary, End-Of-Record (EOR), and Terminal-Type Telnet options. The complete list of 5250 terminal types is maintained in the Assigned Numbers RFC and includes the following:

IBM-5555-C01	24 x 80 Double-Byte Character Set color display
IBM-5555-B01	24 x 80 Double-Byte Character Set (DBCS)
IBM-3477-FC	27 x 132 color display
IBM-3477-FG	27 x 132 monochrome display
IBM-3180-2	27 x 132 monochrome display
IBM-3179-2	24 x 80 color display
IBM-3196-A1	24 x 80 monochrome display
IBM-5292-2	24 x 80 color display
IBM-5291-1	24 x 80 monochrome display
IBM-5251-11	24 x 80 monochrome display

An example of a typical negotiation process to establish 5250 mode of operation is shown below. In this example, the server initiates the negotiation by sending the DO TERMINAL-TYPE request.

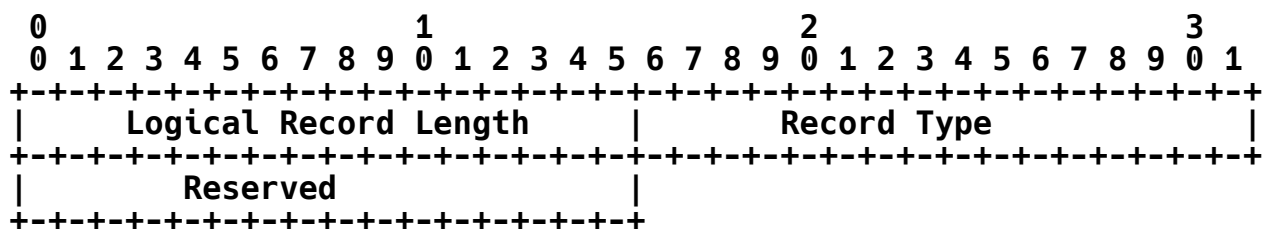
```
Server: IAC DO TERMINAL-TYPE
Client: IAC WILL TERMINAL-TYPE
Server: IAC SB TERMINAL-TYPE SEND IAC SE
Client: IAC SB TERMINAL-TYPE IS IBM-5251-11 IAC SE
(The client has specified its terminal-type is an IBM-5251-11)
```

```
Server: IAC DO END-OF-RECORD
Client: IAC WILL END-OF-RECORD
Server: IAC WILL END-OF-RECORD
Client: IAC DO END-OF-RECORD
(The server and client have both agreed to transmit EORs)
```

```
Server: IAC DO TRANSMIT-BINARY
Client: IAC WILL TRANSMIT-BINARY
Server: IAC WILL TRANSMIT-BINARY
Client: IAC DO TRANSMIT-BINARY
(The server and client have both agreed to binary transmission)
```

### 3. Data Stream Format

The actual data stream that is exchanged between the client and server is composed of a header followed by the 5250 work station data stream. For information about the 5250 work station data stream refer to the IBM 5250 Information Display System, Functions Reference Manual (SA21-9247). The header which prefixes the 5250 data stream was originally designed for the 5250 Display Station Pass-Through (DSPT) application. 5250 DSPT is an application similar to Telnet which runs on the IBM AS/400, System/36, and System/38 over an SNA network. This header is designed to be variable in length and is composed of two parts. The first, fixed part is always 6 octets long and has the following format:



Logical Record Length: 16 bits

This field indicates the length, in octets, of this logical record including the header length. The length is calculated BEFORE

doubling any IAC characters in the data stream. The length does not include the <IAC><EOR> that is appended to the end of the data stream to mark the end of this logical record. The length is specified with the most significant octet first. For example, a length of 36 (decimal) would be specified as '0024'X.

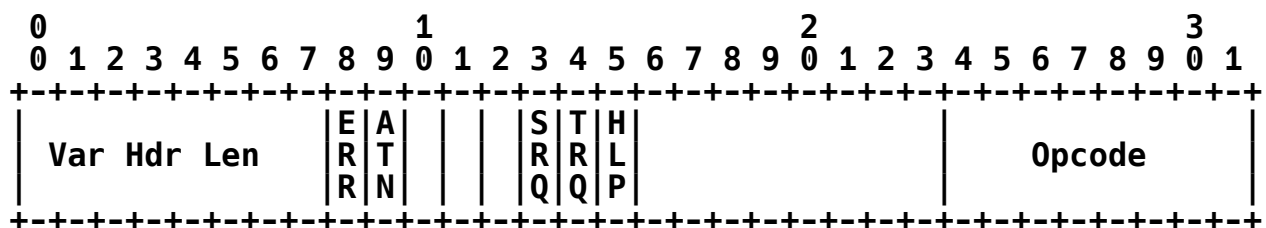
#### Record Type: 16 bits

This field indicates the SNA record type. It should always be set to '12A0'X to indicate the General Data Stream (GDS) record type.

#### Reserved: 16 bits

This field is currently not used.

The second part of the header is designed to be variable in length. The length of this variable part is specified in the first octet. Currently this portion of the header will always be 4 octets long and has the following format:



#### Var Hdr Len: 8 bits

The length, in octets, of the variable portion of the header. Currently this is always '04'X.

#### Flags: 16 bits

Bit 0:	ERR	This bit is set to indicate a data stream output error. The negative response code is sent as data following the opcode field.
Bit 1:	ATN	This bit is set to indicate that the 5250 attention key was pressed.
Bits 2-4:	*	These bits are reserved (set to zero).
Bit 5:	SRQ	This bit is set to indicate that the 5250 System Request key was pressed.
Bit 6:	TRQ	This bit is set to indicate that the 5250 Test Request key was pressed.
Bit 7:	HLP	This bit is set to indicate the Help in Error State function. The error code is sent as data following the header and is

a four digit packed decimal number. For example, an error code of '0005'X indicates the operator attempted to type in an area of the display that is not enabled for input. These bits are reserved (set to zero).

Bits 8-15: \*

Opcode: 8 bits

This field contains the operation code. It is set to indicate the type of operation requested by the sender. The following are the valid values:

'00'X: No Operation  
 '01'X: Invite Operation  
 '02'X: Output Only  
 '03'X: Put/Get Operation  
 '04'X: Save Screen Operation  
 '05'X: Restore Screen Operation  
 '06'X: Read Immediate Operation  
 '07'X: Reserved  
 '08'X: Read Screen Operation  
 '09'X: Reserved  
 '0A'X: Cancel Invite Operation  
 '0B'X: Turn On Message Light  
 '0C'X: Turn Off Message Light

The actual 5250 work station data stream will immediately follow the opcode field in the header and will be terminated by the <IAC><EOR> pair. For some operations the header will be immediately followed by an <IAC><EOR> without any 5250 work station data stream in between. For example, the following request to turn on the message light could be sent by the server:

```

000A 12A0 0000 0400 000B FFEF
|      |      |      |      |
|      |      |      |      | End Of Record marker
|      |      |      |      | Opcode = Turn On Message Light ('0B'X)
|      |      |      |      | Flags = '0000'X
|      |      |      |      | Variable Header Length = '04'X
|      |      |      |      | Reserved - Set to '0000'X
|      |      |      |      | Record Type = General Data Stream ('12A0'X)
|      |      |      |      | Logical Record Length = '000A'X for this record

```

In this example the requested operation is indicated by the opcode and there is no associated work station data stream.

#### 4. Data Flow Examples

The following examples illustrate the flow of data between the client and server for some of the more common operations. These examples are intended to show the order in which the logical records are sent between the client and server and the content of those records. The hex representation of the records which are exchanged between the client and server is shown. The way in which a client implements the various operations will differ between implementations and those details are not discussed here. In these examples, when the value of a field is dependent on the length of the screen data for a particular logical record, it will be represented as 'LLLL'.

##### 4.1 Query Device Example

A Query command may be sent by the server system in order to determine the attributes of the device it is talking to. When a client receives a Query command, it must send the Query Reply back to the server. See section 5 for a complete description of the format for the Query command and Query Reply.

Server: Sends Write Structured Field Query command.	001112A0 00000400 000304F3 0005D970 00FFEF
Client: Responds with a Query Reply, in this case, for a 3180-2.	004712A0 00000400 00000000 88003AD9 70800600 01030000 00000000 00000000 00000000 00000001 F3F1F8F0 F0F0F202 00000061 50000100 00000018 11000000 00000000 000000FF EF

##### 4.2 Cancel Invite Example

The server will send a Cancel Invite when it needs to reverse the normal flow direction. When a client receives a Cancel Invite, it should reply with a Cancel Invite and not send any user data until the server has once again "invited" the work station. A work station is said to be "invited" when the server has sent a read command to the client. The Cancel Invite flow is as follows:

Server: Sends header with the Opcode = Cancel Invite.	000A12A0 00000400 000AFFEF
Client: Sends header with the Opcode = Cancel Invite	000A12A0 00000400 000AFFEF

to indicate that the  
work station is no  
longer invited.

#### 4.3 System Request Example

The 5250 System Request operation is invoked when a client wants to interrupt the server job to perform some function. The typical scenario would be for a user to press the system request key, or whatever key is mapped to a system request key, which would cause the client Telnet to initiate the following flow:

Client: Sends header with the       000A12A0 00000404 0000FFEF  
System Request bit set.

Note: It is possible for a client to include user data in this record following the header. This data would be interpreted by the server as an option to be selected from the system request menu. If this were the case, the server would not send the system request menu and the flow would continue based on the option selected. For this example, the client does not send any user data and the flow would continue as follows:

Server: Sends header with the       000A12A0 00000400 000AFFEF  
Opcode = Cancel Invite.

Client: Sends header with the       000A12A0 00000400 000AFFEF  
Opcode = Cancel Invite  
to indicate that the  
work station is no  
longer invited.

Server: Sends Save (Immediate)       000C12A0 00000400 00040402 FFEF  
command with  
Opcode = Save Screen.

Client: Sends the screen image       LLLL12A0 00000400 00040412  
to be saved.       <Screen Image> FFEF

Server: Sends System Request       LLLL12A0 00000400 0003  
menu with       <System Request Menu> FFEF  
Opcode = Put/Get.

Client: Sends User Input to       LLLL12A0 00000400 0000  
the Sys Req menu.       <User Input> FFEF

Note: What happens next will depend on the System Request option selected by the user. After any System Request processing has

completed, the server will continue with the following restore operation:

Server: Sends the saved LLLL12A0 00000400 00050412  
screen to be restored, <Saved Screen> FFEF  
Opcode = Restore Screen.

(No reply is necessary from the client)

Server: Sends Read Modified 000E12A0 00000400 00010452 0000FFEF  
Data Tag (MDT) command,  
opcode = Invite.

At this point the client would "invite" the workstation and enter the state that it was in prior to the System Request key being hit.

## 5. 5250 Data Stream Enhancements

This section is intended to be used as an addendum to the IBM 5250 Information Display System, Functions Reference Manual. Described here are enhancements to the 5250 data stream which are not yet documented in the current version of that manual. The current version of that manual at the time of this writing has the IBM publication number SA21-9247-6. Also described in this section are corrections to erroneous information contained in SA21-9247-6. Listed below are the specific corrections and enhancements, with an approximate page number reference to the above manual.

### 5.1 Errors or Inconsistencies in SA21-9247-6

The Insert Cursor (IC) order on pages 2-136 and 2-137 is incorrectly listed with a value of Hex 03; the correct value is Hex 13.

On page 2-137, the listed "Restrictions" for the Insert Cursor, Repeat to Address, and Set Buffer Address orders should be updated to describe how Row and Column values must be valid for the current display screen size (either 24 x 80 or 27 x 132).

### 5.2 Enhancements to Existing 5250 Data Stream Commands/Orders

A new flag is added to the second byte of the Control Character on page 2-40. This flag is used to specify whether the cursor should be moved or not moved at the end of the Write to Display processing. Bit 1 of the second byte, which was previously reserved, will now be used for this flag. If bit 1 is a 0, the cursor continues to be moved to the system IC address on a Lock-to-Unlock keyboard transition. If bit 1 is a 1, the cursor is not moved.

A new Field Control Word (FCW) will be added on page 2-65 to indicate an entry field contains transparent data. This means the entry field contents are sent from the display screen directly to the host at read time with no formatting. Therefore, an entry field can contain any values (Hex 00 to Hex FF). A transparent field is indicated by a Hex 84xx FCW, where xx is any value. Note: unpredictable results will occur if a field is defined as both signed numeric and a transparent field.

The Read Immediate, Read Input Fields, and Read MDT Fields commands have been enhanced to include support for transparent fields (page 2-5). If a transparent FCW is found for an input field, the field data is not formatted (for example, nulls are not converted to blanks).

The restriction listed for the Set Buffer Address (SBA) order (page 2-138) on the column address equal to zero is no longer always the case. A reference to Start of Field (SF) row 1/column 1 field support should be made. A note should be added in SF to describe Row 1/Column 1 field support. A Row 1/Column 1 field is defined by a SBA of row 1/column 0, followed by an SF. For a Row 1/Column 1 input field, the first input-capable position is row 1/column 1. If the SF defines an input field, the screen attribute is not allowed to be nondisplay. Writing of the screen attribute is suppressed for a Row 1/Column 1 field and the attribute discarded.

### 5.3 New 5250 Data Stream Commands/Orders

The Read MDT Fields Alternate input command has been added. It is the same as the Read MDT Fields command except:

- The command is indicated by a X'82'
- Leading and embedded nulls within the field remain as nulls

The Read MDT Fields Immediate Alternate input command has been added. It is the same as the Read MDT Fields Alternate command except:

- The command is indicated by a X'83'
- The command is an immediate read command like Read Immediate; therefore, no control characters follow the command byte, field data is returned immediately, and the aid code is X'00'.

The Move Cursor order (MC) has been added (page 2-137). The MC order moves the cursor to the location specified by the two bytes following the order. Byte 1 gives the row address and byte 2 gives the column address. The MC order is useful when the cursor is to be moved without affecting the system IC address. The MC order is unaffected by the Write to Display control character values including the "Leave



Cursor" flag (CC1 bit 1). If more than one MC or IC are found in the data stream, the cursor will move to the address specified in the last MC or IC.

#### Restrictions:

A parameter error will be posted when:

- There are fewer than two bytes following the order.
- The row address is zero or greater than the number of rows on the display screen.
- The column address is zero or greater than the number of columns on the display screen.

#### Format:

Move Cursor Order	Byte 1	Byte 2
X'14'	Row Address	Column Address

#### Results:

The address specified by the MC order is used to move the cursor when the Write to Display is completed.

The Transparent Data order (TD) has been added (page 2-137). The TD order is followed by two length bytes and transparent data. The transparent data is written to the display screen at the current display address; any values (Hex 00 to Hex FF) are allowed in the transparent data. All length values are valid as long as the end of the display screen is not overwritten.

#### Restrictions:

A parameter error will be posted when:

- There are fewer than two bytes following the order.
- There are fewer bytes in the data stream than specified in the length field.
- Attempting to write beyond the end of the display screen.

#### Format:

TD Order	Bytes 1 and 2	Bytes 3 to ?
X'10'	Length of transparent data (not counting length bytes)	Transparent data

**Results:**

The transparent data is written to the display.

The Query command is a new input command (page 2-5) and is used by the server to obtain information on the functional capabilities of the client 5250 display. When the client receives a Query command, the client sends a Query Reply describing its capabilities back to the server.

The Query command must follow an Escape ('04'X) and Write Structured Field command ('F3'X). The format of the Query command is as follows:

Byte	Value	Description
=====	=====	=====
0-1	X'0005'	Length of command
2	X'D9'	Command Class
3	X'70'	Command Type - Query
4	X'00'	Flag Byte
Bit 0:	B'0'	- Query Command
Bit 1-7:		- Reserved (set to zero)

The format of the Query Reply is as follows:

Byte	Value	Description
=====	=====	=====
0-1	X'0000'	Cursor Row/Column (set to zero)
2	X'88'	Inbound Write Structured Field Aid
3-4	X'003A'	Length of Query Reply
5	X'D9'	Command Class
6	X'70'	Command Type - Query
7	X'80'	Flag Byte
Bit 0:	B'1'	- Query Reply
Bit 1-7:		- Reserved (set to zero)
8-9		Controller Hardware Class
	X'0001'	- Local Twinax Controller
	X'0061'	- Local ASCII Controller
	X'0101'	- SDLC/X.21/X.25 Twinax Controller (5394 emulating a 5294)
	X'0103'	- SDLC/X.21/X.25 Twinax Controller (5394)
	X'0200'	- PC DOS non-DBCS WSF
	X'0300'	- OS/2 non-DBCS WSF
	X'0400'	- PC DOS DBCS WSF
	X'0500'	- OS/2 DBCS WSF
	X'0600'	- Other WSF or any other 5250 Emulator
10-12		Controller Code Level
	X'010300'	- For example, Version 1 Rel 3.0
13-28	X'00'	Reserved (set to zero)

29		Device Type
	X'01'	- 5250 Display or 5250 Emulation
30-33	C'cccc'	Device Type (e.g. 3180 for 3180 Mod 2)
34-36	C'ccc'	Device Model (e.g. 002 for 3180 Mod 2)
37		Keyboard ID
	X'02'	- Standard Keyboard
	X'82'	- G Keyboard
38	X'00'	Extended Keyboard ID
39	X'00'	Reserved
40-43	X'xxxxxxxx'	Display Serial Number
44-45		Maximum number of input fields
	X'0100'	- Typically = 256 input fields
46-48	X'00'	Reserved (set to zero)
49-53		Controller/Display Capability
Bit 0-1:	B'00'	- No Row 1/Col 1 support
	B'01'	- Row 1/Col 1 support
Bit 2:	B'0'	- No Read MDT Alternate Command support
	B'1'	- Read MDT Alternate Command support
Bit 3:	B'0'	- Display does not have PA1/PA2 support
	B'1'	- Display does have PA1/PA2 support
Bit 4:	B'0'	- Display does not have PA3 support
	B'1'	- Display does have PA3 support
Bit 5:	B'0'	- Display does not have Cursor Select support
	B'1'	- Display does have Cursor Select support
Bit 6:	B'0'	- Display does not have Move Cursor Order support
	B'1'	- Display does have Move Cursor Order support
Bit 7:	B'0'	- No Read MDT Immediate Alt Command support
	B'1'	- Read MDT Immediate Alt Command support
50		
Bit 0-3:	B'0001'	- 24 x 80 Screen Size
	B'0011'	- Capable of 24 x 80 and 27 x 132
Bit 4:	B'0'	- No light pen support
	B'1'	- Light pen support
Bit 5:	B'0'	- No Mag Stripe Reader support
	B'1'	- Mag Stripe Reader support
Bit 6-7:	B'00'	- Mono display
	B'01'	- 5292/3179 style color, including color PCs
51	X'00'	- Reserved
52		
Bit 0-2:	B'000'	- No Double Byte Character Set (DBCS) capability
	B'001'	- Presentation screen DBCS capability only
Bit 3-7:	B'00000'	- Reserved
53		
Bit 0-2:	B'000'	- No graphics capability
	B'001'	- 5292-2 style graphics
Bit 3-7:	B'00000'	- Reserved
54-60	X'00'	Reserved (set to zero)

## 6. References

1. IBM, "IBM 5250 Information Display System, Functions Reference Manual", SA21-9247-6, March 1987.
2. Postel, J. and J. Reynolds, "Telnet Protocol Specification", RFC 854, USC/Information Sciences Institute, May 1983.
3. Postel, J. and J. Reynolds, "Telnet Option Specifications", RFC 855, USC/Information Sciences Institute, May 1983.
4. Postel, J. and J. Reynolds, "Telnet Binary Transmission", RFC 856, USC/Information Sciences Institute, May 1983.
5. VanBokkeln, J., "Telnet Terminal-Type Option", RFC 1091, FTP Software, Inc., February 1989.
6. Postel, J. and J. Reynolds, "Telnet End of Record Option", RFC 885, USC/Information Sciences Institute, December 1983.

## Security Considerations

Security issues are not discussed in this memo.

## Author's Address

Paul Chmielewski  
IBM Corporation  
Highway 52 and 37 Street North West  
Rochester, Minnesota 55901

Phone: (507) 253-6315

E-Mail: paulc@rchland.iinus1.ibm.com