TEXAS INSTRUMENTS

Improving Man's Effectiveness Through Electronics

Model 990 Computer

DX10 3780/2780 Emulator User's Guide

MANUAL NO. 946289-9701
ORIGINAL ISSUE 1 APRIL 1978
REVISED 15 FEBRUARY 1979

OLD

Digital Systems Group

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES DESTROY SUPERSEDED PAGES

Note: The portion of the text affected by the changes is indicated by a vertical bar in the outer margins of the page.

Model 990 Computer DX10 3780/2780 Emulator User's Guide (946289-9701)

Original Issue . . . . . .1 April 1978
Revised . . . . . .15 February 1979 (ECN 419633)

Total number of pages in this publication is 122 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Cover	0	E-1 - E-6	0		
Effective Pages	0	F-1 - F-6	0		
iii - viii	0	G-1 - G-6	0		
1-0	0	H-1 - H-2	0		
1-1 - 1-6	0	Index-1 - Index-8	0		
2-1 - 2-14	0	User's Response	0		
3-1 - 3-10	0	Business Reply	0		
4-1 - 4-38	0	Cover Blank	0		
A-1 - A-2	0	Cover	0		
B-1 - B-2	0				
C-1 - C-4	0				
D-1 - D-6	0				

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

APPENDIXES

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

GENERAL DESCRIPTION

1.1 INTRODUCTION

An emulator is software or hardware designed to allow programs or routines running under emulator control to appear to be running on the emulated computer system. The DX10 3780/2780 Emulators enable the 990/10 using the DX10 operating system to communicate and exchange files efficiently with another computer over telephone lines (leased or switched, point-to-point). The DX10 3780 and DX10 2780 Emulators emulate the IBM 3780 and 2780 Data Communication Terminals, respectively, so that data sent to or received from the Emulators seems as though it were sent to or received from the IBM 3780 or 2780. This enables the DX10 system, emulating the IBM 3780 or 2780, to communicate and exchange files with a host computer employing the IBM 3780 or 2780 protocol, with an IBM 3780 or 2780, or with another emulator of the IBM 3780 or 2780 (including another DX10 system equipped with the DX10 3780 or 2780 Emulator). Figure 1-1 indicates some of the possible applications of the Emulator.

NOTE

A DX10 2780 Emulator can communicate only with hosts, terminals, or other emulators that employ the IBM 2780 protocol. The DX10 3780 Emulator can communicate only with hosts, terminals, or other emulators that employ the IBM 3780 protocol.

Since the Emulators have similar features and operating instructions, the DX10 3780 and the DX10 2780 are addressed in this manual as a single emulator. Differences in their features and operation are called to your attention through discussions designated as "2780 Emulator Users Only" or "3780 Emulator Users Only," and the emulator is referred to as "the 2780 Emulator" or "the 3780 Emulator" throughout the discussion. Otherwise, the 2780 Emulator and the 3780 Emulator are referred to as "Emulator" and any discussion not referenced as aforesaid applies to both Emulators.

1.2 IBM 3780 AND 2780 DATA COMMUNICATION TERMINALS

The IBM 3780 and 2780 Data Communication Terminals are nonprogrammable terminals that provide their users with remote file transfer capabilities when connected to an IBM host system. The IBM 3780 and 2780 Data Communication Terminals being emulated consist of a controller, a card reader, a line printer, and optionally, a card punch.

These Terminals use a card reader as their input device and may use a card punch and/or line printer as their output devices. Files transferred to and from the IBM systems are restricted to these devices.

1.3 THE EMULATOR CONFIGURATION

The Emulator runs on the 990/10 under the DX10 operating system. In addition to the minimum hardware requirements described in Appendix A, this system may be configured to include the following:

- Disk media (including flexible diskette)
- Video display terminals
- Hard-copy data terminals (ASR/KSR)

- Transmission of data
- Reception of data
- Use of special features

The Emulator maintains a log of the transactions that occur during the communication session. The log can be maintained on the operator console, a file, or a device, depending on the environment.

When all desired transmissions and receptions have taken place, the Emulator is terminated and the communication session ends.

1.6 HARDWARE AND SOFTWARE REQUIREMENTS

Appendix A and Appendix B outline the hardware and software requirements, respectively, for the operation of the Emulator.

1.7 INSTALLATION

Standard or user-defined default values are set for use during Emulator operation at the time the Emulator is installed. These values determine the defaults for:

- Timeout lengths
- Retry counts
- Features enabled or disabled
- Block sizes
- Record sizes

If the standard default values are not appropriate for your application, they can be modified during installation. The installation procedures and a table of the standard values are in Appendix C.

1.8 FEATURES AND CAPABILITIES

The following briefly describes the IBM features and capabilities emulated by the Emulator, the features and capabilities unique to the Emulator, and the IBM features and capabilities not emulated by the Emulator.

1.8.1 EMULATED IBM FEATURES AND CAPABILITIES.

- Code Structure — EBCDIC.
- Data Link Control — Binary Synchronous Communications (BSC) Procedures control point-to-point data exchanges.
- Switched Network Control — The Emulator can automatically answer or disconnect when the communication line is attached to a switched (dial-up) network and is attached with an appropriate auto-answer modem.
- Conversational Mode (3780 Emulator Users Only) — The Emulator can accept data from a host system as a positive acknowledgement to the last block of a transmitted message.
- Extended Enquiry Retry — The Emulator can retry a bid for the communication line an infinite number of times.

- Dual Communications Interface — The IBM 2780 can communicate on an alternate line if its primary communication line becomes disabled.
- Terminal Testing — The IBM 3780 can perform a test, online or offline, of its communication ability.
- Synchronous Clock — The IBM 2780 and 3780 can use data sets having no clocking signals since the synchronous clock feature provides them with an internal clock.

SECTION II

FUNCTIONAL DESCRIPTION

2.1 INTRODUCTION

This section describes the IBM features and capabilities emulated by the Emulator and the features unique to the Emulator.

2.2 EMULATED FEATURES AND CAPABILITIES

The following paragraphs describe the features and capabilities of the IBM 3780 and 2780 that are emulated by the Emulator.

2.2.1 COMMUNICATION LINES. The Emulator supports leased, switched, or equivalent privately-owned telephone lines for its communications.

2.2.2 CODE STRUCTURE. The IBM 3780 and 2780 support EBCDIC code and, optionally, ASCII code. The IBM 2780 also optionally supports the Six-Bit Transcode. The DX10 system uses the ASCII code structure, but the Binary Synchronous Communications (BSC) procedures emulated support EBCDIC, not ASCII. The Emulator translates ASCII files to EBCDIC prior to transmission. If communicating with another DX10 3780/2780 Emulator, the receiving Emulator translates the files back to ASCII following reception (unless do not translate is specified using the binary specification, refer to paragraph 2.2.13, Binary Data Transmissions/Reception). An ASCII to EBCDIC, EBCDIC to ASCII translation table is provided in Appendix G.

2.2.3 DATA LINK CONTROL. The line communications performed by the Emulator follow the same Binary Synchronous Communications procedure for point-to-point exchanges which is used by the IBM 3780 and 2780 Data Communication Terminals. Multipoint data link control, which enables the IBM 3780 and 2780 to operate on a communication line with multiple stations, is not supported by the Emulator. The BSC procedure, which is transparent to the user, provides transmission checking and basic control of the data link. The data transmission is performed in half-duplex mode at line speeds up to 9600 bps. The actual speed is determined by the user-selected modem.

Table 2-1 lists the data link control characters supported by the Emulator and describes their most common usage. Additional information about the Binary Synchronous Communications procedures and data link control characters may be found in Appendix F.

Table 2-1. Data Link Control Characters

Control Character	Description
SYN	Precedes data being transmitted to provide synchronization.
STX	Indicates the beginning of a transmission block.
ITB (IUS)	Breaks long transmissions into convenient sizes for error checking.
ETB	Indicates the end of one transmission block of a multiple transmission block message.

Table 2-1. Data Link Control Characters (Continued)

Control Character	Description
ETX	Indicates the end of the last transmission block of a message.
EOT	Indicates the end of a transmission.
ENQ	Indicates that the station sending the ENQ has data to transmit.
ACK0/1	Indicates the receiving station has received the transmitted block of data successfully.
NAK	Indicates the receiving station has received the transmitted block of data unsuccessfully, i.e., the transmission had errors in it.
DLE	Combines with other characters to provide additional data link control characters.
WACK	Indicates the receiving station has received the transmitted block of data successfully and requests that the transmitting station temporarily delay before its next transmission.
RVI	Indicates the receiving station has received the transmitted block of data successfully and requests that the transmitting station stop transmitting so that the receiving station may transmit priority data.
TTD	Indicates the transmitting station has data to transmit, is temporarily not ready to transmit it, but does not wish to give up the line.

2.2.4 SWITCHED NETWORK CONTROL. As with the IBM 3780 and 2780, the Emulator is capable of automatic answer and disconnect when the line is attached to a switched (dial-up) network and is equipped with an appropriate auto-answer modem.

The automatic answer and disconnect feature enables the Emulator to do the following:

- Automatically answer an incoming call without operator intervention.
- Receive and execute a disconnect command from a remote station (a DLE EOT sequence is sent by the remote station).
- Automatically disconnect when the connection is not active. The Emulator automatically disconnects the connection after a 20-second timeout if no meaningful activity has occurred. (This feature is selectable during installation.) If at any time during the timeout meaningful activity takes place, the timeout restarts with a new 20-second time period. The disconnection can be accomplished sooner through the use of the Disconnect (DISC) command if desired.
- Cancel a transmission initiated from an input device if the transmission is initiated prior to beginning the disconnect. This is accomplished by the transmitting station issuing an ABORT COM command or physically breaking the connection, or by the modem sensing a disconnect from the remote station after the transmission is initiated.

2.2.5 CONVERSATIONAL MODE (3780 EMULATOR USERS ONLY). This feature enables the 3780 Emulator to accept data from the host system as a positive acknowledgement (a conversational reply) to the last data block of a transmitted message. The data must begin with an STX character or a DLE STX if in transparent mode. Refer to paragraph 2.2.12, EBCDIC Transparency, for a description of transparent mode. The output device (file) must be ready for this feature to be operable.

2.2.6 EXTENDED ENQUIRY (ENQ) RETRY. The extended ENQ retry feature permits the IBM 3780 and 2780 to retry an ENQ fifteen times before timing out with an error condition. The Emulator default retry number is fifteen retries; this may be increased or decreased during installation.

2.2.7 INPUT/OUTPUT DEVICES. The IBM 3780 and 2780 have one input device: the card reader. The Emulator allows input from the card reader, from other input devices, and from sequential files. For transmissions, the Emulator formats the data so that it appears to the remote stations to have come from a card reader.

The IBM 3780 and 2780 have two output devices: the line printer and the card punch. The Emulator does not support the card punch but may direct received card punch data to any local output device or to a sequential file. The Emulator can format data so that it can be punched by the receiving station. Line printer data may be received and directed to a line printer, to any local output device, or to a sequential disk file.

Cassette tape may be used for Emulator input or output only if the 733 ASR is not also the Emulator operator console. Input or output using slow devices could cause error messages on the host system console.

2.2.8 HORIZONTAL AND VERTICAL FORMAT CONTROL. The Emulator supports the horizontal format control and most of the vertical format control (VFC) characters used by the IBM 3780 and 2780. Table 2-2 indicates the horizontal format and VFC characters used by the IBM 3780 and 2780 and supported by the Emulator. Because certain of the printers used with the 990/10 system do not use a carriage control tape, the Emulator translates the "Skip to channel n" VFC characters to carriage returns (CRs) and line feeds (LFs). While this translation may not provide the user with a line-for-line representation of the print file as it was created, it does provide a format similar to the one desired.

Usually, printer data that is received by the Emulator and directed to a disk file has VFC information stripped from the file so the data can be processed by standard DX10 utilities. The VFC feature allows the operator to specify that VFC information be retained on the files. This allows the user to maintain print files on disk for later printing without sacrificing format control. This feature is enabled or disabled using the Vertical Format Control (VFC) command described in paragraph 4.5.1.13 or by specifying enabled or disabled as the default value during installation.

Files are output with single spacing when directed to devices other than a line printer or disk file.

Table 2-2. IBM 3780 and 2780 and 3780/2780 Emulator Horizontal and Vertical Format Control Characters

EBCDIC Code	IBM Printer Operation	TI Printer Operation	TI Character Sequence
NL (15)	Start a new line, terminate HT, and execute any pending ESCs	Same	CR
IRS (1E)	Start a new line, do not terminate HT, and execute any pending ESCs	Same	IRS
VT (0B)	Vertical Tab — skip to channel 2	Same	VT
FF (0C)	Skip to heading line (channel 1)	Same	FF
LF (25)	Start a new line (Line Feed)	Same	LF
HT (05)	Horizontally tabbed, space-compressed data, when received, is to be expanded to fit set horizontal tab format	Spaces to correct column	
ESC/	Single Space	Carriage Return/ Line Feed	CR LF
ESC S	Double Space	Carriage Return/ 2 Line Feeds	CR LF LF
ESC T	Triple Space	Carriage Return/ 3 Line Feeds	CR LF LF LF
ESC A	Skip to channel 1	Carriage Return/ Form Feed	CR FF
ESC B	Skip to channel 2	Carriage Return/ Vertical Tab	CR VT
ESC C	Skip to channel 3	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC D	Skip to channel 4	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC E	Skip to channel 5	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC F	Skip to channel 6	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC G	Skip to channel 7	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF

Table 2-2. IBM 3780 and 2780 and 3780/2780 Emulator Horizontal and Vertical Format Control Characters (Continued)

EBCDIC Code	IBM Printer Operation	TI Printer Operation	TI Character Sequence
ESC H	Skip to channel 8	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC I*	Skip to channel 9	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC J*	Skip to channel 10	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC K*	Skip to channel 11	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC L*	Skip to channel 12	Carriage Return/ 6 Line Feeds	CR LF LF LF LF LF LF
ESC M	Space Suppress	Carriage Return	CR

*Not supported by the 2780 Emulator.

2.2.9 HOME MODE. The Emulator does not operate in home mode, but home mode operations, e.g., offline card reader to printer operations, can be performed by the DX10 copy concatenate utilities.

2.2.10 SPACE COMPRESSION/EXPANSION (3780 EMULATOR USERS ONLY). This feature allows the 3780 Emulator, prior to transmission, to replace strings of spaces (2 to 63) in data with an IGS character followed by a space-count character indicating the number of spaces removed. The spaces are reinstated upon reception and the IGS character is deleted. For additional groups of space characters (at least two), an additional IGS character and space-count character must be inserted. This feature improves transmission efficiency. This feature should be used in an Emulator-to-host system environment only when compression/expansion is supported by the host system. This feature is enabled or disabled using the Compression/Expansion (CE) command described in paragraph 4.5.1.2, or by specifying enabled or disabled as the default value during installation.

2.2.11 END OF MEDIA (2780 EMULATOR USERS ONLY). The End of Media (EM) feature allows the 2780 Emulator to place an end of record indicator at the end of the data in transmitted records. When this character is read by an IBM 2780 or another emulator that supports EM, further reading of the record is prevented. The 2780 Emulator uses end of media to truncate trailing blanks from input records. This allows faster transmission of data since trailing blanks are not transmitted. The end of media feature is enabled or disabled using the End of Media (EM) command described in paragraph 4.5.1.3 or by specifying enabled or disabled as the default value during installation.

2.2.12 EBCDIC TRANSPARENCY. In normal (nontransparent) data transmissions, certain EBCDIC characters perform control functions (refer to Appendix F for an explanation of the control functions). To send these characters, e.g., STX or ETX, as a part of a message without having them perform their control function, the data transfer must be performed in transparent mode. In transparent mode, only the control character DLE (Data Link Escape) is recognized by the receiver. All other transmitted characters are accepted as data. For the other control characters to be recognized as control characters, they must be preceded by a DLE character. To transmit the DLE character transparently within the message block, it must be preceded by another DLE character.

A transparent transmission is signaled to the receiving station by the characters DLE STX preceding the message. These characters are automatically inserted when transparency is enabled using the Transparency (TY) command described in paragraph 4.5.1.12 or is set as the default condition during installation.

For the IBM 3780 and 2780, all 256 eight-bit combinations, including binary data, may be transmitted or received in transparent mode. For the Emulator, transparency enabled means that the entire seven-bit ASCII code set may be transmitted or received. To transmit or receive binary data (or all 256 eight-bit combinations) with the Emulator, the special identifier # (pound sign) must precede the file pathname or text string. This special identifier indicates to the Emulator that it should not translate from ASCII to EBCDIC as it normally would. Refer to paragraph 2.2.13, Binary Data Transmissions/Receptions, for further information.

2.2.13 BINARY DATA TRANSMISSIONS/RECEPTIONS. In a 990 to 990 communication environment, the Emulator can transmit and receive binary data using the Send Data (S), Send Data with Device Selection (SS), Receive Printer Data (RPR), and Receive Punch Data (RPU) commands (these commands are described further in Section IV, paragraphs 4.5.3.1, 4.5.3.2, 4.5.4.1, and 4.5.4.2, respectively). The binary specification may also be used on nonbinary files. The binary specification in this case means do not translate this file from ASCII to EBCDIC and back as the Emulator normally would. This enables the Emulator to transmit and receive all 256 eight-bit combinations (the entire EBCDIC code). Binary specification also allows the transfer of DX10 program files.

Binary is specified using a # (pound sign). The pound sign precedes each pathname being transmitted or received. For further information about sending and receiving binary files, refer to paragraphs 4.5.3 and 4.5.4, respectively.

2.2.14 COMPONENT SELECTION. The component selection feature permits the Emulator and the IBM 3780 and 2780 to have their output device(s) selected by the transmitting station. Selection is made via a component selection code sent by the transmitting station. If no component selection is made, received data is directed to the printer.

The 3780 Emulator has two selection codes for the card punch. Either a DC2 or DC3 may be selected for the card punch at installation time. The selection code chosen for the local station's card punch should match the card punch selection code of the remote station.

The 3780 Emulator selection code for the printer is DC1.

The 3780 Emulator selection code is valid only if it is received as the first text characters of the first block of text, whether that block is:

- Received after transmitting a positive response to a line bid.
- The first block of a conversational reply.
- The first block following an ETX block.

If the selection code is not transmitted in this manner, it is handled as data.

The 2780 Emulator card punch selection code is ESC4. The printer selection code is any of the valid EBCDIC printer ESC sequences shown in table 2-2.

The 2780 Emulator selection code must be received as the first two characters of the first record of a nontransparent transmission block or as the first two characters following the first STX in a nontransparent block of data. The selection code may be included in other records also. To use the selection code for transparent data, first send the selection code alone in nontransparent mode, then, send the subsequent blocks as transparent data. The selection code will remain in effect until another selection is made in nontransparent mode or until an EOT (End of Transmission) is sent.

The Emulator, in addition to receiving the selection codes, transmits them. The selection code is automatically transmitted in the appropriate manner when the Send with Device Selection (SS) command is used. The card punch or printer is selected through the operator's response to the DEVICE: prompt. When transmitting to an IBM host, SS should not be used since the selection code is not understood.

The output device selected must be ready. If it is not, an EOT character is transmitted as a response to the block of data.

2.2.15 AUDIBLE ALARM. The audible alarm feature on the IBM 3780 and 2780 alerts the operator, with an audible alarm, that the terminal requires attention. Indicator lights on the terminal indicate the reason for the alarm. When the Emulator requires attention, messages that correspond to the meaning of the IBM 3780 and 2780 indicator lights are logged on the operator console or in a log file. Situations that prompt these messages are indicated in table 2-3.

Table 2-3. Emulator Audible Alarm Messages

NOTES

The messages "*** TRANSMIT COMPLETE nn IIIII RRRRR 00000 pathname/text string" and "*** RPRINT COMPLETE nn IIIII RRRRR 00000 pathname" have been shortened to the portion significant to the table. For the complete message, refer to Appendix E, table E-1.

When the message "*** RPRINT COMPLETE nn IIIII RRRRR 00000 pathname" is indicated, the message "*** RPUNCH COMPLETE nn IIIII RRRRR 00000 pathname" is also valid.

IBM 3780 and 2780 Audible Alarm Conditions	DX10 3780/2780 Emulator Messages
At a transmitting terminal:	
An attempt to transmit during which the communications line cannot be secured.	*** BID ERROR
An incorrect response to a redundancy check causes the transmission to abort.	*** TRANSMIT COMPLETE 56
A card reader or buffer error causes the transmission to abort.	*** I/O ERROR XX pathname[1] *** TRANSMIT COMPLETE 51
An EOT character is received following a transmission of a block of data indicating that the receiving station is aborting the transmission.	*** TRANSMIT COMPLETE 51

Table 2-3. Emulator Audible Alarm Messages (Continued)

IBM 3780 and 2780 Audible Alarm Conditions	DX10 3780/2780 Emulator Messages
A disconnect takes place while operating in auto-answer mode on a switched line.	
—If disconnect takes place while data is being transferred:	*** TRANSMIT COMPLETE 18 *** LINE DISCONNECTED
—If disconnect takes place while the line is idle:	*** LINE DISCONNECTED
At normal end of job.	*** TRANSMIT COMPLETE 00
At the receiving terminal:	
A printer or buffer error aborts the transmission. (An EOT transmitted in response to an abort condition.)	*** I/O ERROR XX pathname[1] *** RPRINT COMPLETE 51
The BEL character is received.	*** BELL RECEIVED
The buffer has been overrun and the cyclic redundancy characters compare (i.e., the transmission is otherwise without errors).	*** RPRINT COMPLETE 00 IIIII RRRRR 00000[2]
A disconnect takes place while operating in auto-answer mode on a switched line.	
—If disconnect takes place while data is being transferred:	*** RPRINT COMPLETE 18 *** LINE DISCONNECTED
—If disconnect takes place while the line is idle:	*** LINE DISCONNECTED
A component is selected which is not ready.	*** I/O ERROR XX pathname[1]

[1]The unidentified data in this message (i.e., IIIII, RRRRR, XX, etc.) is explained in Appendix E.
[2]This condition is indicated by a number in the 00000 field (the number of records truncated).

2.2.16 VARIABLE RECORD LENGTH. The Emulator and the IBM 3780 and 2780 can in certain situations transmit and receive variable length records. The transmitting and receiving environments determine those situations. When deciding on the lengths to specify for the Send Record Length command described in paragraph 4.5.1.10, the Receive Punch Record Length command described in paragraph 4.5.1.8, and the Receive Print Record Length command described in paragraph 4.5.1.5, the following must be taken into consideration:

- Some hosts do not support End of Media (EM).
- Some hosts do not support Compression/Expansion (CE).

- The send record length specified should consider the receiving terminal's output capabilities and should be equal to or less than the receiver's output maximums.
- The receive record lengths (punch or print) specified should consider the transmitting terminal's capabilities and should be equal to or greater than the expected transmitted record lengths.
- For transmission/reception of fixed length records, both terminals should agree on the record length.

Table 2-4 indicates the environments in which the Emulator transmits and receives and the record lengths allowable for those environments.

Table 2-4. Transmission/Reception of Variable Length Records

990/2780 TO 990/2780

Transparency OFF

Transmit: With EM ON:

Variable length records
Maximum record length = Send Record Length (SRL)

With EM OFF:

Fixed length records
Record length = SRL

Receive: With EM ON/OFF:

Variable length records
Maximum record length = Receive Print Record Length (RPRRL) for printer
Receive Punch Record Length (RPURL) for punch

Transparency ON

Transmit: With EM ON/OFF:

Fixed length records
Record length = SRL

Receive: With EM ON/OFF:

Fixed length records
Record length = SRL

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

Transparency ON — Binary

Transmit: With EM ON/OFF:

Variable length records (single record blocks)
Maximum length = SRL (record length will equal length read from device)

Receive: With EM ON/OFF:

Variable length records (single record blocks)
Maximum record length = RPRRL for printer
RPURL for punch
Record length = record length received

990/3780 TO 990/3780

Transparency OFF

Transmit:
Variable length records
Maximum record length = SRL

Receive:
Variable length records
Maximum record length = RPRRL for printer
RPURL for punch

Transparency ON

Transmit:
Fixed length records
Record length = SRL

Receive:
Fixed length records
Record length = RPRRL for printer
RPURL for punch

Transparency ON — Binary

Transmit:
Variable length records (single record blocks)
Maximum record length = SRL
Record length = record length read from input device

Receive:
Variable length records (single record blocks)
Maximum record length = RPRRL for printer
RPURL for punch

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

990/2780 TO IBM 2780

Transparency OFF

Transmit: With EM ON:

Variable length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

With EM OFF:

Fixed length records
Record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: With EM ON/OFF:

Variable length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

Fixed length records
Record length = 80 characters
RPRRL — printer
RPURL — punch

Transparency ON

Transmit: With EM ON/OFF:

Fixed length records
Record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: With EM ON/OFF:

Fixed length records
Record length = 80 characters

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

990/3780 TO IBM 3780

Transparency OFF

Transmit: Variable length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: Variable length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

Transparency ON

Transmit: Fixed length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: Fixed length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

990/2780 TO IBM HOST

Transparency OFF

Transmit: With EM ON:

Variable length records
Maximum record length = SRL
80 characters

With EM OFF:

Fixed length records
Record length = SRL
80 characters

Receive: With EM ON/OFF:

Variable length records
Maximum record length = 80 characters — punch
132 characters — printer
RPRRL — printer
RPURL — punch

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

990/2780 TO IBM 2780

Transparency OFF

Transmit: With EM ON:

Variable length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

With EM OFF:

Fixed length records
Record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: With EM ON/OFF:

Variable length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

Fixed length records
Record length = 80 characters
RPRRL — printer
RPURL — punch

Transparency ON

Transmit: With EM ON/OFF:

Fixed length records
Record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: With EM ON/OFF:

Fixed length records
Record length = 80 characters

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

990/3780 TO IBM 3780

Transparency OFF

Transmit: Variable length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: Variable length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

Transparency ON

Transmit: Fixed length records
Maximum record length = SRL
80 characters — punch
80/120/144 characters — printer

Receive: Fixed length records
Maximum record length = 80 characters
RPRRL — printer
RPURL — punch

990/2780 TO IBM HOST

Transparency OFF

Transmit: With EM ON:

Variable length records
Maximum record length = SRL
80 characters

With EM OFF:

Fixed length records
Record length = SRL
80 characters

Receive: With EM ON/OFF:

Variable length records
Maximum record length = 80 characters — punch
132 characters — printer
RPRRL — printer
RPURL — punch

Table 2-4. Transmission/Reception of Variable Length Records (Continued)

Transparency ON

Transmit: With EM ON/OFF:

Fixed length records
Record length = 80 characters

Receive: With EM ON/OFF:

Fixed length records
Record length = 80 characters — punch
132 characters — printer
RPURL — punch
RPRRL — printer

990/3780 TO IBM HOST

Transparency OFF

Transmit: Variable length records
Maximum record length = 80 characters
SRL

Receive: Variable length records
Maximum record length = 80 characters — punch
132 characters — printer
RPURL — punch
RPRRL — printer

Transparency ON

Transmit: Fixed length records
Record length = 80 characters
SRL

Receive: Fixed length records
Record length = 80 characters — punch
132 characters — printer
RPURL — punch
RPRRL — printer

2.2.17 MULTIPLE RECORD TRANSMISSION. The 2780 Emulator allows the specification of the number of records to be placed in a block for transmission or reception. When communicating with an IBM 2780 with the multiple record transmission feature installed or with an IBM host, up to seven records (or not more than 400 characters) may be transmitted or received. The 3780 Emulator allows as many records as will fit in a block to be transmitted or received. When communicating with an IBM 3780 or with an IBM host, as many records as will fit in a block (512 characters) may be transmitted or received.

2.3 ADDITIONAL EMULATOR FEATURES

In addition to the IBM features and capabilities that the Emulator supports, the Emulator provides three additional features: the Receive Pathname feature, the Send Request feature, and the Automatic Call Unit (ACU) feature.

2.3.1 RECEIVE PATHNAME FEATURE. In a 990 to 990 communication environment, the destination pathname for a transmitted file may be specified by the transmitting station using the Receive Pathname feature. A four-character recognition code, assigned during installation, specifies to the receiving station that the pathname following the recognition code will be the receive pathname for the next file transmitted. The recognition code and the destination pathname may be transmitted as:

- A text string
- The first record of a data file
- The only record of a data file

For information about the operation of the Receive Pathname feature, refer to paragraph 4.5.4.3.

2.3.2 SEND REQUEST FEATURE. In a 990 to 990 communication environment, the local station can send a request to the remote station that it send a specified file to the local station. This capability is the Send Request feature. Essentially, the local station sends a Send Data (S) or Send Data with Device Selection (SS) command to the remote station. The remote station then executes that S or SS command transmitting the file specified in the command to the local station. A four-character recognition code, assigned during installation, signals the receiving station that the data following the code is an S or SS command to be executed. The Send Request send commands are somewhat more than the send commands described in paragraphs 4.5.3.1 and 4.5.3.2. The Send Request commands include a specification of EBCDIC transparency on or off, end of media (2780) or compression/expansion (3780) on or off, and the send record length.

The recognition code and Send Request send command can be transmitted as:

- A text string
- The first record of a data file
- The only record of a data file

For information about the operation of the Send Request feature, refer to paragraph 4.5.3.3.

2.3.3 AUTOMATIC CALL UNIT (ACU) FEATURE. The ACU is a printed circuit board that may be installed on the 990 system. The ACU interfaces with the modem to allow the operator to place a call to a remote station without dialing a telephone. With an ACU installed, the operator issues a Call (CALL) command, the system prompts for a telephone number, the operator enters the telephone number, and the call is automatically made.

Two types of ACUs may be used with the 990 system. The internal ACU is a Texas Instruments ACU used with a Texas Instruments modem. The external ACU is a user-supplied ACU used with a user-supplied modem. The external ACU interfaces with the 990 using the Texas Instruments External Auto-Call Unit Interface.

These two ACUs operate alike except for tandem dialing. Tandem dialing allows the user to dial an intermediate number to access an outside line (get a second dial tone) to make the telephone call. Tandem dialing is supported for the internal ACU. Tandem dialing is not supported for the external ACU.

SECTION III

FUNCTIONAL ENVIRONMENT

3.1 INTRODUCTION

All communication and file exchanges using the Emulator take place in an environment referred to as a communication session. A communication session begins when the system operator executes the appropriate Emulator (refer to paragraph 4.2, Emulator Activation). During Emulator activation, the DX10 System Command Interpreter (SCI) issues prompts to determine the following:

- Command source (COMMAND ID:)
- Log destination (LOG ID:)
- Communication line (COMM DEVICE ID:)
- Automatic Call Unit (ACU) Communication Register Unit (CRU) Address, if so equipped (ACU CRU ADDR:)

The command source may be one of the following:

- Video display terminal or a hard-copy display terminal operated by the operator (operator mode)
- Input device or sequential or relative record file (command file mode)
- Intertask message queue (intertask message queue mode)

The log destination may be one of the following:

- Video display terminal
- Hard-copy display terminal
- Output device
- Sequential or relative record file
- Intertask message queue

The communication line is one of the following:

- CM01 when only one communication line is attached to the system
- CM01 through CMnn when multiple communication lines are attached to the system

The ACU CRU ADDR: prompt is only displayed if an ACU is installed. The CRU address is determined during installation.

During the communication session, the Emulator issues command prompts to which the operator or the designated command source responds. These responses initiate the Emulator transactions. The Emulator performs each transaction and then issues messages that indicate the status of the transaction (i.e., completed successfully, completed unsuccessfully error encountered, etc.). While these transactions are taking place, the Emulator is keeping a log of the transactions. The log is displayed on or written to the log destination specified during activation. The communication session is ended by terminating the Emulator.

3.2 OPERATIONAL MODES

The Emulator transactions are initiated by Emulator commands. The Emulator commands may be issued by an operator, a sequential file, an input device or an intertask message queue. During activation, the operator must specify what the command source will be. The response given to the COMMAND ID: prompt specifies the command source for the session. This command source is considered the primary command source and determines the mode of operation for the session. The mode may be:

- Operator Mode — The operator enters commands on a VDT or hard-copy display terminal.
- Command File Mode — The Emulator gets its commands from a sequential file or input device.
- Intertask Message Queue Mode — The Emulator gets its commands from an intertask message queue.

During the operation of the Emulator, the primary command source can issue a Command File (CF) command specifying that further commands will be issued by a secondary command source, a command file. In operator mode or intertask message queue mode, issuing this command opens the command file specified which then becomes the new command source. The session then continues in command file mode. In command file mode, issuing this command closes the active command file and executes the new one.

3.2.1 OPERATOR MODE. The operator uses a video display terminal or hard-copy display terminal to issue commands (Section IV), respond to command prompts, and respond to console messages. The prompts, responses, and Emulator messages are logged on the operator console. While in operator mode, the primary log destination must be the video display terminal or hard-copy display terminal used to issue commands. Refer to paragraph 4.4.1 for further information about Emulator operation in operator mode.

3.2.2 COMMAND FILE MODE. A command file is a prepared set of responses to the command prompts issued from a specified pathname (device or file). A command file can be the primary or secondary command source. A command file is specified as the primary command source when a command file pathname is entered as the response to the COMMAND ID: prompt during Emulator activation. Once a command file is specified as the primary command source, the command file commands direct the session until the Emulator is terminated or a secondary command source is specified. The Emulator also operates in command file mode when a command file is specified as the secondary command source. A command file becomes the secondary command source when the primary command source issues a Command File (CF) command. Refer to paragraph 4.4.2 for a discussion of Emulator operation in command file mode.

3.2.3 INTERTASK MESSAGE QUEUE MODE. DX10 tasks running on the DX10 system may be specified as the primary command source through the use of intertask message queues. The commands issued by the DX10 task are records representing a response to a command prompt. The

commands are placed by the DX10 task into an intertask message queue. The task identifies the queue with a hexadecimal number between 00 and FF. This queue identification number is assigned as the Emulator's COMMAND ID: during Emulator activation.

3.3 RECORD OF THE SESSION

A continuous record of the communication session is made while the Emulator is operating. This record is called a log and includes Emulator prompts, operator responses, command file commands, intertask message queue commands, and Emulator message output. The log can be output to the following:

- Video display terminal
- Hard-copy display terminal
- Output device
- Sequential or relative record file
- Intertask message queue

While in operator mode, the record of the session can only be output on the operator terminal. If the operator terminal is a hard copy device, a permanent record of the session may be maintained. If the operator terminal is a VDT, the record of the session is restricted to what is currently displayed on the screen. Previously displayed information is not retrievable. In command file mode, the record of the session can be output to any file, to any available output device (including the operator terminal), or to any intertask message queue specified. In intertask message queue mode, the record can be output to any file, to any available output device (including the operator terminal), or to an intertask message queue.

When using the Send Request feature, the record of the session is shared between the two Emulators according to their participation in the data exchange (e.g., the Emulator transmitting the request logs the prompts, responses, and messages associated with the transmission; the Emulator receiving the request and then making the requested transmission logs the prompts, responses, and messages associated with the reception and the requested transmission). Each Emulator outputs the logged information to the file, device, or queue that it is currently using for log output.

3.3.1 PRIMARY LOG DESTINATION. The primary log destination is determined by the operator's response to the LOG ID: prompt. The primary log destination may be one of the following:

- Video display terminal
- Hard-copy display terminal
- Output device
- Sequential or relative record file
- Intertask message queue

A log directed to the video display terminal cannot be printed. A log directed to an intertask message queue can be printed only if a task is programmed to read the queue and print it.

3.3.2 SECONDARY LOG DESTINATION. A secondary log destination can be specified by the primary command source using the Log File (LF) command. When the LF command is issued, log information stops going to the primary log destination and begins going to the secondary log destination. When the primary command source is the operator terminal, a secondary log destination may be specified, but no log information will go to the secondary log destination until a command file is specified.

3.4 COMMAND PROMPTS

Command prompts are helpful messages, displayed on the operator display terminal or logged in a file, in a queue, or on a device, that guide you in supplying the information required for the operation of the Emulator. After the Emulator is activated, the Emulator issues a COMMAND: prompt. The command source enters a command in response. If the command has parameters or options associated with it, a prompt is issued requesting the parameter or option. In operator mode, the command prompts are displayed on the operator display terminal. The parameters or options available with each command are described in Section IV. Table 3-1 lists the command prompts and their meanings.

Table 3-1. Command Prompts

Prompt	Meaning/User Action
COMMAND:	The operating system is ready to accept an Emulator command.
DEVICE:	The command just issued requires one of these parameters: PR (printer) PU (punch)
PATHNAME:	The command just issued requires a legal DX10 pathname or synonym as its response (refer to paragraph 4.3.2). The Send Data (S) and Send Data with Device Selection (SS) commands, described in paragraphs 4.5.3.1 and 4.5.3.2, respectively, also allow a text string as a response.
LENGTH:	The command just issued requires that a record or block length be specified.
NUMBER:	The command just issued requires a number as its parameter.
OPTION:	The command just issued requires that the user specify an option or indicate whether a feature is to be enabled or disabled. The options are listed with the commands in Section IV.

3.5 COMMANDS

From Emulator activation through Emulator termination, all of the Emulator transactions consist of issuing commands and responding to prompts or messages. The commands can be divided into the following categories:

- Change default commands
- Connect commands
- Send data commands

- Receive commands
- Command file commands
- Disconnect commands
- Miscellaneous commands

The change default commands are used to change installation parameters. A default value is set during installation that specifies:

- The block length
- The receive print record length
- The receive punch record length
- The send record length
- The records per block (for 2780)
- The default pathname for the printer
- The default pathname for the punch
- The features enabled or disabled

The change default commands enable the operator to change those default values for the current communication session. These commands are often issued prior to transmitting or receiving since they set the specifications for the data.

The connect commands place a call or answer a call to establish a communication line connection. They may be issued whenever you are ready to transmit or receive data.

The send data commands are used to send data. One send command can send as many files as buffer space will allow.

The receive commands are issued each time a reception is expected. The receive commands specify the pathname in which the data will be received.

The command file commands are issued to enter command file mode and to aid in the execution of a command file. The Command File (CF) command and the Log File (LF) command are used by the operator to specify that a command file begin executing its commands and a log file begin receiving the log of those transactions. The IF and WAIT commands are issued by the command file to give the command file versatility. A command file may be executed anytime during a communication session as a temporary command source or may be specified during activation as the primary command source for the session.

The disconnect commands disconnect the telephone connection. The Abort and Terminate Task commands can also terminate the Emulator.

The miscellaneous commands are issued to reset the communication line and to display the statistics for the session.

3.6 DATA TRANSMISSION

A data transmission is initiated by any of the following: the operator, a command file command, an intertask message queue command, or a send request (which issues an S or SS command specifying the pathname(s) of the file(s) to be sent). Each transmission may include as many files or text strings as there are queue buffers available at the time. When multiple files are transmitted using one Send command, the files are concatenated and sent as one file. Any number of transmissions may be generated (with sufficient queue buffers), since they are queued and handled one at a time.

Using the pathname specified in the command, the Emulator "opens" the pathname and starts the transfer of the data to an input buffer. From the device/file input buffer, the data is translated (or not translated if specified), compressed (if specified for 3780), and moved to one of two communication blocking buffers. When the communication blocking buffer is filled, the data is queued for transfer to the communication line. The output data transfer continues in this fashion until the end-of-file (EOF) is reached, an irrecoverable error occurs, or a processor interrupt is received from the host system. Figure 3-1 illustrates the buffering of a data transmission.

3.6.1 INTERRUPTED TRANSMISSIONS. Occasionally, transmissions are interrupted. The interruption may be initiated by the transmitting station or the receiving station through commands, physical disconnections, communication link errors, or system errors. The command which interrupts a transmission is the Abort command. When ABORT TASK or ABORT COM is issued by the transmitting or receiving terminal, any transmissions in progress are aborted and deleted from the queues.

The host system can interrupt a transmitting station's transmissions by responding to a transmission with an RVI. This tells the transmitting station to stop transmitting so that the host system can transmit priority data. When the host completes its transmission, the transmitting station resumes its transmissions where it left off.

A physical disconnection initiated by either terminal is treated as a communication link error. When a communication link error occurs:

- During a transmission, the transmission is aborted, deleted from the queue, and the next transmission in the queue is acted upon.
- Prior to beginning a transmission, the transmission is left in the queue and is the next transmission transmitted.

When system errors occur (usually input/output errors or open errors) during a transmission or prior to beginning a transmission, the transmission is aborted, deleted from the queue, and the next transmission in the queue is sent.

3.7 DATA RECEPTION

Data reception is initiated by the remote station bidding for the communication link. During communication sessions, the Emulator looks for incoming data when it is not transmitting. When received, the bid is accepted and data is received from the line into the alternating communication buffers and transferred from there to an intermediate buffer for system output (figure 3-2). Data recognition is validated using the transmitted Cyclic Redundancy Check (CRC) characters (refer to Appendix F), and retransmissions are automatically requested if the path is erroneous. Using the pathname specified in the Receive Printer Data (RPR) or Receive Punch Data (RPU) command, the Emulator "opens" the receive pathname and starts the transfer of data from the buffer into the pathname.

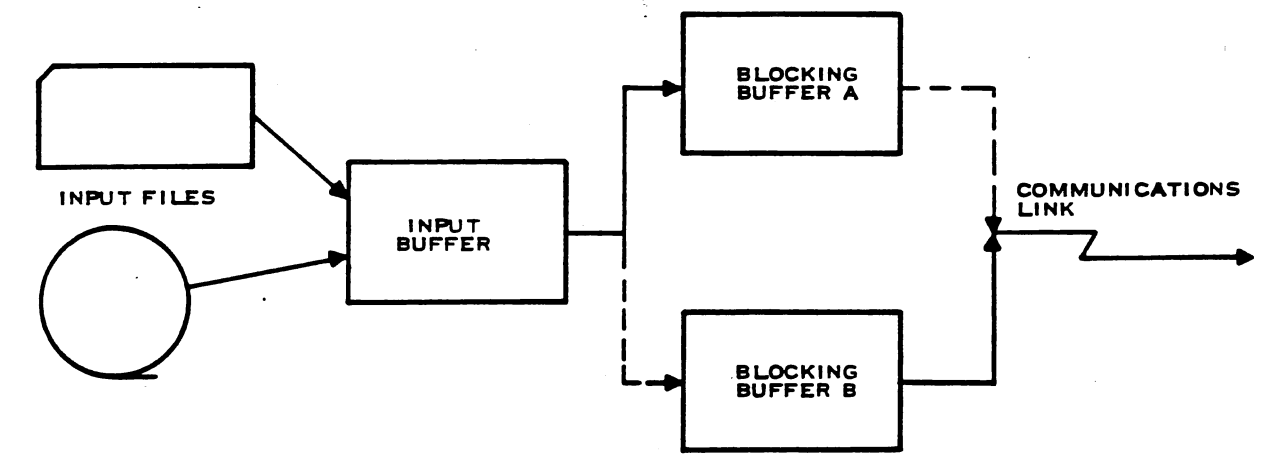


Figure 3-1. Data Transmission Buffering

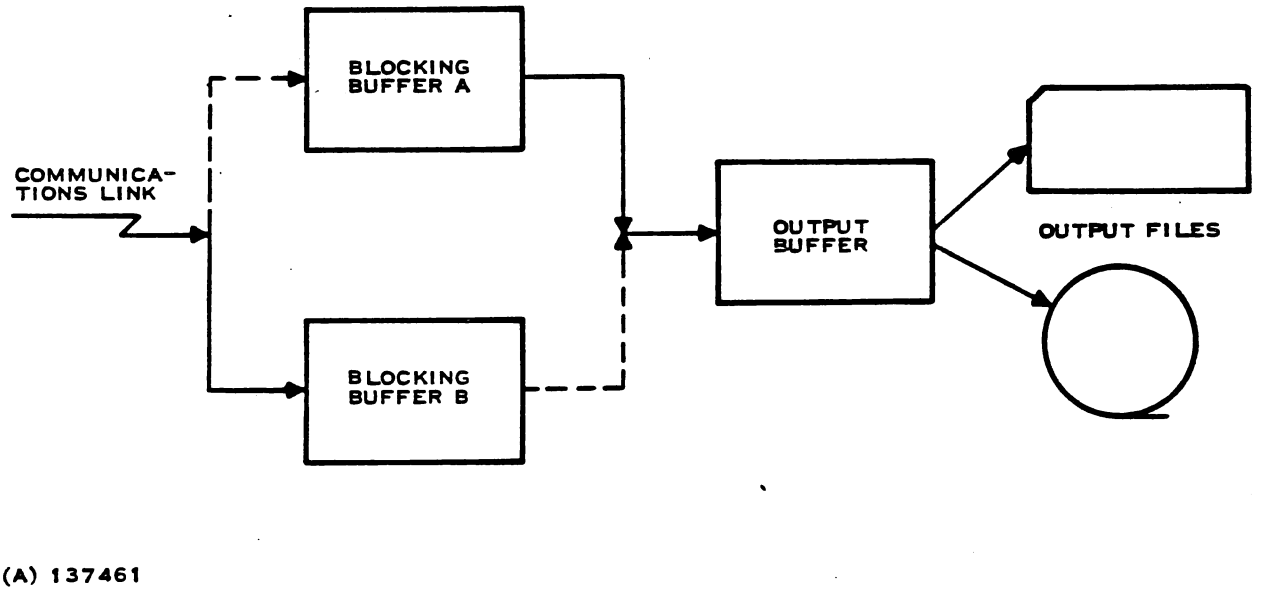


Figure 3-2. Data Reception Buffering

3.7.1 INTERRUPTED RECEPTIONS. Occasionally, a reception is interrupted by the transmitting station or the receiving station through commands, physical disconnections, communication link errors, or system errors. The command that can interrupt a reception is the Abort command. When ABORT TASK or ABORT COM is issued by either the transmitting or receiving station, any transactions in progress are aborted and deleted from the queues.

A physical disconnection initiated by either station is treated as a communication link error. When a communication link error occurs, the data received prior to the error is placed in the pathname specified and the reception is deleted from the queue. The data that is not received prior to the error is lost.

When a system error occurs (usually an input/output error or an open error):

- While trying to open the receive pathname, the data goes to the next pathname in the queue. If the error occurred while trying to open the default pathname, the remote station is notified, and any data is discarded.

- After opening the receive pathname, the data received prior to the error is placed in the pathname, the remote station is notified of the error, and the remainder of the data is discarded.

3.8 SESSION MESSAGES

The Emulator can display or log two types of session messages: transaction and statistics. Transaction messages are displayed or logged automatically in response to Emulator transactions. Statistics are displayed or logged using the Statistics command.

3.8.1 TRANSACTION MESSAGES. The transaction messages indicate the following:

- Successful completion of transactions

- Status of transactions

- Error conditions

The messages usually consist of a short phrase that describes the transaction, its status, or an error condition.

Several messages have variable information. These messages are formatted using one of three possible formats as follows:

- A short phrase describing the transaction followed by the pathname of the file concerned:

  RPNAME COMMAND pathname

- A short phrase describing the error followed by an error code and the pathname of the file concerned:

  OPEN ERROR XX pathname

- A short phrase describing the transaction and its status followed by:

  Completion/error code

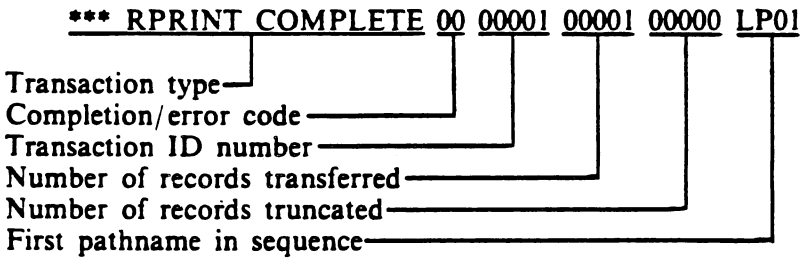
  Transaction identification number

  Number of records transferred

  Number of records truncated (i.e., the number of records in the transfer that were longer than the maximum specified record length)

  Pathname of the file transferred (the first pathname for a transaction with multiple pathnames)

Example:



The transaction messages are listed in table E-1 of Appendix E. The completion/error codes are described in table E-2 of Appendix E.

3.8.2 STATISTICS. Statistics are compiled internally as the Emulator operates. To list these statistics, the Statistics (ST) command is used. The Statistics command has options that enable the operator to display or log the following sets of statistics:

- Current Emulator conditions—the transactions in progress, the status of the communication line, the status of features (enabled or disabled), the record and block length values for the session.

- Communication link statistics since the most recent connection—the number of BIDs, BID accepts, TTDs, NAKs, WACKs, ENQs, EOTs, and ACKs received and transmitted; the number of Conversational Replies, RVIs, and BID overrides received; the number of times the BID, NAK, ENQ, and IDLE thresholds have been exceeded.

- Transactions on the send queue—the transaction ID and the pathname for each transaction on the send queue.

- Transactions of the receive print queue—the transaction ID and the pathname for each transaction on the receive print queue.

- Transactions on the receive punch queue—the transaction ID and the pathname for each transaction on the receive punch queue.

- Transactions on all queues—the transaction ID and the pathname for each transaction for each queue (presented by queue).
- All statistics—a listing of all the statistical information compiled.

The Statistics (ST) command and the Emulator statistics are described further in paragraph 4.5.7.2.

SECTION IV

EMULATOR OPERATION

4.1 INTRODUCTION

Operation of the Emulator begins with Emulator activation. The activation defines the operating environment. When activation is complete, communication and file exchanges may begin. Communication and file exchanges are accomplished using the Emulator commands. The commands may be issued by an operator in operator mode, by a command file in command file mode, or by an intertask message queue in intertask message queue mode.

The user-supplied data in the examples in this section is italicized to distinguish user input from Emulator input.

4.2 EMULATOR ACTIVATION

The Emulator may be installed with or without an Automatic Call Unit (ACU). The activation procedure for an Emulator with an ACU includes all of the activation steps for an Emulator without an ACU but also includes special steps concerning the ACU.

4.2.1 EMULATOR ACTIVATION WITHOUT ACU. An Emulator without an ACU is loaded and activated from a station by entering the appropriate command, X3780 or X2780, following the command mode indicator.

[ ]*X2780* or [ ]*X3780*

After the command is entered, the appropriate menu of prompts is presented:

```
EXECUTE 2780 EMULATOR          or    EXECUTE 3780 EMULATOR
   COMMAND ID:  ME                        COMMAND ID:  ME
         LOG ID:  ME                            LOG ID:  ME
COMM DEVICE ID:                        COMM DEVICE ID:
```

This menu enables the user to enter the source of the command input (operator, command file, intertask message queue), the destination of the log for the session (operator terminal, command file, intertask message queue), and the identification of the communication line to be used. The command source ID and the log destination ID may be any device identifier or file pathname allowed by DX10 (including synonyms) or intertask message queue identification numbers. The command source ID and log destination ID should not be the same unless the operation of the Emulator is in operator mode. In operator mode, the command source and the log destination must be the operator terminal. ME is the default for the COMMAND ID: and LOG ID: prompts. Using the default specifies that the primary command source and the primary log destination for this session is the terminal displaying the menu.

The following are examples of completed menus.

```
EXECUTE 2780 EMULATOR          or    EXECUTE 3780 EMULATOR
   COMMAND ID:  CR01                      COMMAND ID:  ME
         LOG ID:  LP01                          LOG ID:  ME
COMM DEVICE ID:  CM01                  COMM DEVICE ID:  CM01
```

Subsequent executions of the Emulator will display the menu including the previous response to the COMM DEVICE ID: prompt. If this response is correct, use a carriage return to avoid rekeying.

If an error is made in response to one of the prompts, the system responds with one of the following messages:

COMMAND DEVICE NOT AVAILABLE

LOG DEVICE NOT AVAILABLE

INVALID PARAMETERS

If the responses are all correct, the Emulator is executed and responds with a header of the form:

DX10/XXXX V.R.U. YY.DDD

where

XXXX is 2780 or 3780, whichever is being executed,
V is the version number,
R is the release number,
U is the update level,
YY is the last two numbers of the year of the Emulator release, and
DDD is the Julian date within the year of the Emulator release.

This is followed by the Emulator COMMAND: prompt.

4.2.2 EMULATOR ACTIVATION WITH ACU. An Emulator with an ACU is loaded and activated from a station by entering the appropriate command X3780A or X2780A following the command mode indicator.

[ ]*X2780A* or [ ]*X3780A*

After the command is entered, the appropriate menu of prompts is presented:

```
EXECUTE 2780 EMULATOR WITH ACU        or   EXECUTE 3780 EMULATOR WITH ACU
        COMMAND ID:  ME                            COMMAND ID:  ME
           LOG ID:  ME                                LOG ID:  ME
    COMM DEVICE ID:                            COMM DEVICE ID:
      ACU CRU ADDR:                              ACU CRU ADDR:
```

This menu enables the user to enter the source of the command input (operator, command file, intertask message queue), the destination of the log for the session (operator terminal, command file, intertask message queue), the identification of the communication line to be used, and the automatic call unit CRU address. The command source ID and the log destination ID may be any device identifier or file pathname allowed by DX10 (including synonyms) or intertask message queue identification numbers. The command source ID and log destination ID should not be the same unless the operation of the Emulator is in operator mode. In operator mode, the command source and the log destination must be the operator terminal. ME is the default for the COMMAND ID: and LOG ID: prompts. Using this default specifies that the primary command source and the log destination will be the terminal displaying this menu. The ACU CRU address is a hexadecimal number giving the location of the ACU on the CRU bus. If the user's system has only one communication line, the procedure may be edited to automatically pass the CRU address to the task

internally and this response can be eliminated. For systems with more than one ACU, this prompt requires the CRU address of the ACU to be used during operation of the Emulator. The correct CRU addresses for your ACUs are assigned during installation.

The following menus are shown as they would appear when completed by the operator.

EXECUTE 2780 EMULATOR WITH ACU

COMMAND ID: *CR01*
LOG ID: *LP01*
COMM DEVICE ID: *CM01*
ACR CRU ADDR: *>FFFF*

or

EXECUTE 3780 EMULATOR WITH ACU

COMMAND ID: ME
LOG ID: ME
COMM DEVICE ID: *CM01*
ACR CRU ADDR: *>FFFF*

Subsequent executions of the Emulator will display the menu with the previous response to the COMM DEVICE ID: prompt. If this response is correct, use a carriage return to avoid rekeying.

If an error is made responding to any of the prompts, the system responds with one of the following messages:

COMM DEVICE NOT AVAILABLE

LOG DEVICE NOT AVAILABLE

INVALID PARAMETERS

If the responses are all correct, the Emulator is executed and responds with a header of the form:

DX10/XXXX V.R.U. YY.DDD

where

XXXX is 2780 or 3780, whichever is being executed,
V is the version number,
R is the release number,
U is the update level,
YY is the last two numbers of the year of the Emulator release, and
DDD is the Julian date within the year of the Emulator release.

This is followed by the Emulator COMMAND: prompt.

4.3 EMULATOR COMMAND FORMAT

The format of the Emulator commands provides an easy-to-use command sequence. Operator commands consist of prompts followed by operator responses. The first prompt is always COMMAND:. This may be followed by additional prompts that vary depending on the command.

Commands may be issued in general format or condensed format. The general format begins with the COMMAND: prompt. The response to the COMMAND: prompt is a command followed by a carriage return. The next prompt for this command, if any, is then issued. Each prompt issued is answered with an appropriate response and completed with a carriage return. The command is terminated by responding with a valid response to the last prompt issued for the command, by following the last response with an asterisk and carriage return, or by responding to a prompt with a

carriage return (null response). When the primary command source is other than an operator, each record is considered a response followed by a carriage return. Two examples of the general format are:

```
COMMAND:   SS                    COMMAND: RPN
DEVICE:    PR                    OPTION:  ON
PATHNAME:  .DIR.FILE
PATHNAME:  return
```

The condensed format of the operator command begins with the COMMAND: prompt. All the parameters for the command are entered following the command. The command and each parameter are followed by asterisks. If the final asterisk is omitted and additional parameters could be entered, the appropriate prompt for the next expected parameter is issued; i.e., the command returns to the general format for the remainder of the command or until a parameter followed by an asterisk is given. If a terminating asterisk is omitted by mistake, respond to the next prompt with a carriage return. If an asterisk terminates a prompt line, no other input for this command will be accepted. For Command File mode and Intertask Message Queue mode, the condensed format is more convenient to use. The condensed format uses only one record per command while the general format uses one record per command and one record for each command parameter. Examples of the condensed format are:

COMMAND: *SS*PR*.DIR.FILE*.DIR.FILE1*

COMMAND: *RPN*ON**

COMMAND: *S*.DIR.FILE**

The terminating asterisk may be omitted when there are no more input parameters, as in the statistics command: ST*ALL* or ST*ALL. The terminating asterisk must be used to terminate the condensed format of the Send Data (S) and Send Data with Device Selection (SS) commands. For the S and SS commands, the terminating asterisk follows the last pathname input. The terminating asterisk must be used to terminate the condensed format of the CALL command if the user wishes to specify manual dial on a system configured with an ACU. In this case, the terminating asterisk follows the command mnemonic CALL. The following are examples of the use of the terminating asterisk.

The following example shows the condensed format terminated with an asterisk indicating that .DIR.FILE1 is the last pathname.

COMMAND: *S*.DIR.FILE*.DIR.FILE1**

An example of the condensed format without a terminating asterisk follows. The asterisk is not necessary because there are no more input parameters.

COMMAND: *RPRRL*100*

The following example shows the condensed format with the terminating asterisk omitted by mistake. The condensed format returns to the general format with its next prompt.

```
COMMAND:   S*.DIR.FILE*.DIR.FILE1
PATHNAME:  return
```

In the following example, an asterisk terminates the CALL command because the user's system is equipped with an ACU, but the user wishes to dial the call manually. The asterisk prohibits displaying the NUMBER: prompt.

COMMAND: *CALL**

In the next example, no terminating asterisk is required since the system is not equipped with an ACU (i.e., no other parameters are allowed). After issuing the CALL command, the operator manually dials the call.

COMMAND: *CALL*

Any operator response to a command or command parameter prompt that contains an ampersand (&) aborts that command. Another COMMAND: prompt is then displayed. An ampersand (&) is accepted in the response to a PATHNAME: prompt of an S or SS command as long as it is enclosed in angle brackets (< >). In most cases an ampersand would not be used in the condensed format since an incorrect response to the COMMAND: prompt can be corrected by backspacing and retyping the response. The condensed format is included for comparison purposes in the examples showing use of the ampersand to abort commands.

In the following example, the operator enters an ampersand to abort the RPRRL command.

General format:

COMMAND: *RPRRL*
LENGTH: *&*
COMMAND: *next command*

Condensed format:

COMMAND: *RPRRL*&*
COMMAND: *next command*

In the next example, the operator enters an ampersand to abort the SS command.

General format:

COMMAND: *SS*
DEVICE: *PR*
PATHNAME: *&*
COMMAND: *next command*

Condensed format:

COMMAND: *SS*PR*.DIR.FILE*&*
COMMAND: *next command*

In the example which follows, the ampersand is part of a text string and does not abort the command.

General format:

COMMAND: S
PATHNAME: *<THIS IS TEXT&>*
PATHNAME: *return*

Condensed format:

COMMAND: *S*<THIS IS TEXT&>**

An example of the ampersand entered outside the text string, thus aborting the command, follows:

General format:

COMMAND: *S*
PATHNAME: *<THIS IS TEXT>&*
COMMAND: *next command*

Condensed format:

COMMAND: *S*<THIS IS TEXT>*&*
COMMAND: *next command*

A ? response to a prompt causes a list of all valid responses to that prompt to be displayed and the prompt to be repeated. A ? response is valid, in general, for any prompt that does not require a number or a pathname as its response. When a question mark is entered in a command in condensed format, the command format returns to the general format for the remainder of the command. The following examples show the use of the question mark as a response.

General format:

COMMAND: *RPN*
OPTION: *?*
ON OFF ?
OPTION: *option*

Condensed format:

COMMAND: *RPN*?**
ON OFF ?
OPTION: *option*

General format:

COMMAND: *ABORT*
OPTION: *?*
COM TASK ID ?
OPTION: *option*

Condensed format:

COMMAND: *ABORT*?**
COM TASK ID ?
OPTION: *option*

If a blank is encountered in the operational portion of a response, the response is terminated at the blank. This allows comments to be written on command records in command files. Blanks may appear within a text string as the response to the PATHNAME: prompt of the S or SS command. These examples show responses containing blanks.

The following is an example of a blank entered in error.

General format:

COMMAND: *RPRR L*
*** INVALID INPUT
COMMAND:

Condensed format:

COMMAND: *RPRRL* 100**
*** INVALID INPUT
COMMAND:

The following example shows blanks within a text string (enclosed in angle brackets).

General format:

COMMAND: *S*
PATHNAME: *<THIS IS TEXT>*
PATHNAME: *return*

Condensed format:

COMMAND: *S*<THIS IS TEXT>**

The following example uses blanks to allow comments to be written on the record.

COMMAND: *S THIS IS THE SEND COMMAND*
PATHNAME: *.DIR.FILE THIS IS MY PATHNAME*
PATHNAME: *return*

4.3.1 PATHNAME RESPONSES. The required response for many of the Emulator commands is a standard DX10 pathname. The pathname of a file is a concatenation of the volume name, the directory levels (excluding VCATALOG) leading to the file, and the file name itself. Components of the pathname are separated by periods. The volume name need not be included if the volume is the system volume. The total length of a pathname must be 48 characters or less. The following are examples of valid pathnames:

Volume name, directory name, directory name, file name:

VOL.DIR.DIR1.FILE

Directory name, directory name, file name (volume will be system disk):

.DIR.DIR1.FILE

Volume name, directory name:

VOL.DIR

Directory name, file name:

.DIR.FILE

A special form of pathname, called the $-option, is used by the Receive Printer Data (RPR) and Receive Punch Data (RPU) commands. Any standard pathname can be used as a $-option pathname by placing a $ at the end of the pathname, e.g.,

Standard Pathname:	**$-option Pathname:**
VOL.DIR.FILE	VOL.DIR.FILE$

During the operation of the Emulator, the $ is assigned a consecutive number, 01 through 99 (or some value less than 99 if the capacity of the device is exceeded), each time a file is received in the queue containing the $-option pathname. In this way, multiple files may be received without having to issue multiple RPR or RPU commands.

For further information about pathnames, refer to the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume II,* part number 946250-9702.

4.3.2 SYNONYM RESPONSES. A string of one or more characters can be defined to substitute for a pathname and used as a response to a prompting message. A synonym can represent an entire pathname or a component of a pathname. Using the pathname VOL.DIRB.DIRC.FILE, synonyms could be assigned as in the following:

Synonym	**Synonym Value**
A	VOL
AB	VOL.DIRB
ABC	VOL.DIRB.DIRC
ABCD	VOL.DIRB.DIRC.FILE

Using these synonyms, the following are valid pathnames:

A.DIRB.DIRC.FILE
AB.DIRC.FILE
ABC.FILE
ABCD

When a synonym is used as the destination pathname of the Receive Pathname feature or as the requested pathname of the Send Request feature, the synonym must be defined at the remote station. Otherwise, synonyms need to be defined only at the station where the synonym is entered.

For further information about synonyms, refer to the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume II,* part number 946250-9702.

4.4 MODES OF OPERATION

The Emulator can operate in three modes:

- Operator Mode
- Command File Mode
- Intertask Message Queue Mode.

The mode of operation determines the commands used and the organization of the session.

4.4.1 OPERATOR MODE OPERATION. In operator mode, the operator issues commands, responds to prompts, and responds to Emulator messages. The commands are issued as they are needed.

The log destination is the operator terminal. If the operator terminal is a hard-copy terminal, a written record of the session can be maintained. Otherwise, the log is only retained as long as the log information is displayed on the video display.

While in operator mode, a command file may be specified. Once specified, the command file becomes the command source for the session. Control is not returned to the operator unless:

- The last command of the command file does not terminate the task.
- A second command file is specified with a null pathname.
- An error occurs in the command file.

4.4.2 COMMAND FILE MODE OPERATION. In command file mode, the commands are issued from a specified sequential or relative record file or input device. Each command in the command file is a record representing one response to a prompt. The format of the response may be either general or condensed (the condensed format is more convenient to use). If the general format is used, each response is one record; null responses are represented by blank records. Each response has a maximum record length of 80 characters.

The log destination while in command file mode may be the operator terminal, a sequential or relative record file, an output device, or an intertask message queue.

Operating in command file mode requires anticipating all of the transactions for the duration of command file mode. The command file must include the same commands that would be used in operator mode. Because the operator cannot change the command file commands while the command file is executing, the Emulator provides two commands that give the command some decision-making capabilities. The IF and WAIT commands are used by command files to provide the command file with some of the versatility of an operator.

The IF command is used in an IF—$ELSE—$END structure to allow the command file to conditionally execute or skip a group of its commands. When the specified condition is true, the Emulator executes all commands until $ELSE is encountered in the command stream, then skips all commands until $END is encountered. If the specified condition is false, the Emulator skips all commands until $ELSE or $END is encountered. $ELSE and $END are entered as commands.

Nested IF commands are not allowed. If a second IF command is encountered anywhere within the original IF—$ELSE—$END structure, it is treated as an invalid command, the message ***INVALID INPUT is logged, and execution continues with the next command.

The WAIT command allows the command file to wait until a condition is satisfied before executing the next command. This is particularly helpful when the next command depends on the previous command for its input. The WAIT will wait for a connection, a disconnection, a queue to become empty, or all queues to become empty. The WAIT command has certain safeguards built into it to prevent the command file from waiting indefinitely for a condition that will never be satisfied. These safeguards are:

- For a wait until connect complete, the WAIT will be executed only if the Emulator is in a condition in which connection can be accomplished, e.g., a CALL or ANS command has been issued prior to the WAIT command.

- For a wait until queue empty, the WAIT will be executed only if a connection has been established so that it is possible for those queues to go empty.

Once a command file has been executed, the user cannot interrupt the operation of the Emulator (unless he physically disconnects the connection or kills the task). For this reason, it is important that care be exercised in arranging the command file. The following suggestions should help avoid problems.

- Make sure that the directory of a multilevel pathname is available or, if it is not, that a default pathname is specified.

- Make sure that the number of transactions queued prior to issuing a CALL command do not exceed the queue buffer space. If this happens, the Emulator waits for queue buffer space to be available; but without a connection, no buffer space can become available.

A command file may be specified as the primary command source or the secondary command source. The command file operation is almost the same in either case. The primary difference is that as a primary command source the command file can issue a CF command to execute a second command file. The secondary command file issues its commands and then execution returns to the primary command source. When a secondary command file issues a CF command, the secondary command file is closed and the newly specified command file begins execution. When the command file ends, execution returns to the primary command source, not to the secondary command file.

When the command file is the primary command source, it controls the session until an end-of-file (EOF) is read, a TERM or ABORT TASK command is encountered, or a device error occurs in the command chain. When any of these occur, the task terminates.

When the command file is the secondary command source, it controls the session until an end-of-file (EOF) is read, a TERM or ABORT TASK command is encountered, an error occurs in the command chain, or another CF command is encountered. When the command file is terminated by an EOF or an error, Emulator control returns to the primary command source beginning with the command following the CF command. If a CF command is encountered in the active command file, the executing command file is terminated and the new one begins execution, i.e., nested CF commands are not allowed. Issuing a CF command with a null (blank) pathname causes the active command file to be closed and session control to be returned to the primary command source.

4.4.3 INTERTASK MESSAGE QUEUE MODE OPERATION. A DX10 task running on the DX10 system can place Emulator commands into an intertask message queue. The commands issued by the DX10 task are records representing a response to a prompt. The format of these commands may be either general or condensed (the condensed form will usually be more convenient). If the general format is used, each response is one record and null responses are represented by blank records. These commands have a maximum record length of 80 characters. To use intertask message queues, the system must have been generated to include intertask communication and adequate buffers.

The commands are placed by the task into an intertask message queue. The task identifies the queue with a hexadecimal number between 00 and FF. This queue identification number is assigned as the Emulator's COMMAND ID: during Emulator activation.

A second intertask message queue or a log file may be identified to accept log information from the Emulator. Like the command intertask message queue, the log intertask message queue is identified by a hexadecimal number between 00 and FF. This queue should be identified as a separate queue from the command queue. The queue identifier or the log file pathname is assigned as the Emulator's LOG ID: during Emulator activation. The console prompts, the intertask message queue responses, and any messages resulting from the interaction are logged. Using a command intertask message queue does not require using a log intertask message queue. The Emulators can accept commands from an intertask message queue and output log information to a log file, the operator console, or an intertask message queue; and they may accept commands from a file and log to an intertask message queue.

For information about intertask communication, refer to the *Model 990 Computer DX10 Operating System Release 3 Reference Manuals*, part numbers 946250-9701 through 946250-9706.

4.5 EMULATOR COMMANDS

The Emulator commands are grouped according to their function. Within each function, the commands are listed in alphabetical order by command name. The groups are as follows:

- Change Default Commands
- Connect Commands
- Send Data Commands
- Receive Data Commands
- Command File Commands

- Disconnect Commands
- Miscellaneous Commands.

4.5.1 CHANGE DEFAULT COMMANDS. During installation, default values are set for transmission block lengths, send and receive record lengths, records per block (2780), receive print and punch default pathnames, and enabling or disabling features. The change default commands enable the user to change the default values as necessary during the session. The installed defaults are listed in Appendix C.

4.5.1.1 Block Length Command (BL). Use the BL command to specify the block length for transmitted data.

General format:

COMMAND: *BL*
LENGTH: *block length*

Condensed format:

COMMAND: *BL*block length**

Notes:

The block length for the 2780 may be any decimal or hexadecimal number in the decimal range 80 through 400 (>50 through >190). A hexadecimal number must be preceded by the greater-than character (>). For communications with an IBM system, the block length must be 400 or >190.

The block length for the 3780 may be any decimal or hexadecimal number in the decimal range 80 through 512 (>50 through >200). A hexadecimal number must be preceded by the greater-than character (>). For communications with an IBM system, the block length must be 512 or >200.

Examples:

General format:

COMMAND: *BL*
LENGTH: *288*

Condensed format:

COMMAND: *BL*288**

4.5.1.2 Compression/Expansion Command (CE) (3780 Emulator Users Only). Use the Compression/Expansion command to enable or disable the Compression/Expansion feature. This feature removes consecutive space characters from input data before transmission.

General format:

COMMAND: *CE*

OPTION: { *ON* | *OFF* | *?* }

Condensed format:

COMMAND: *CE** { *ON* | *OFF* | *?* } ***

Notes:

If transparency is on (TY=ON), compression/expansion is ignored.

Some host systems do not support compression/expansion; therefore, it is necessary to specify ON/OFF for the 3780 Emulator according to the system with which you are communicating.

Examples:

General format:

COMMAND: *CE*
OPTION: *ON*

COMMAND: *CE*
OPTION: *OFF*

Condensed format:

COMMAND: *CE*ON**

COMMAND: *CE*OFF**

4.5.1.3 End of Media Command (EM) (2780 Emulator Users Only). Use the EM command to enable or disable the End of Media feature. This feature places an end of record indicator at the end of the data in a record. When the indicator is read by the receiving station, no further reading of the record is done. In this way, trailing blanks are truncated. End of media is valid for nontransparent data only.

General format:

COMMAND: *EM*

OPTION: $\left\{\begin{matrix} ON \\ OFF \\ ? \end{matrix}\right\}$

Condensed format:

COMMAND: $EM*\left\{\begin{matrix} ON \\ OFF \\ ? \end{matrix}\right\}*$

Notes:

Some host systems do not support end of media; therefore, it is necessary to specify ON/OFF for the 2780 according to the system with which you are communicating.

Specifying end of media ON or OFF affects transmitted data only; the Emulator operates properly for any state of received data.

If transparency is on (TY=ON), end of media is ignored.

Examples:

General format:

COMMAND: *EM*
OPTION: *ON*

COMMAND: *EM*
OPTION: *OFF*

Condensed format:

COMMAND: *EM*ON**

COMMAND: *EM*OFF**

4.5.1.4 Receive Pathname Command (RPN). Use the RPN command to enable or disable the Receive Pathname feature. With RPN enabled, the Emulator looks for the special Receive Pathname four-character recognition code followed by a pathname. The recognition code is set during installation. With RPN disabled, the Emulator treats the four-character recognition code and pathname as data. The Receive Pathname feature is described further in paragraphs 2.3.1 and 4.5.4.3.

General format:

COMMAND: *RPN*

OPTION: { *ON* | *OFF* | *?* }

Condensed format:

COMMAND: *RPN** { *ON* | *OFF* | *?* } ***

Notes:

The Receive Pathname recognition code used by the transmitting station must be the same as the code used by the receiving station.

Examples:

General format:

COMMAND: *RPN*
OPTION: *ON*

COMMAND: *RPN*
OPTION: *OFF*

Condensed format:

COMMAND: *RPN*ON**

COMMAND: *RPN*OFF**

4.5.1.5 Receive Print Record Length Command (RPRRL). Use the RPRRL command to specify a record length for received line printer data.

General format:

COMMAND: *RPRRL*
LENGTH: *record length*

Condensed format:

COMMAND: *RPRRL*record length**

Notes:

The record length may be any decimal or hexadecimal number in the range from one to the maximum output buffer size specified in the user parameter table. Hexadecimal numbers must be preceded by the greater-than character (>).

Records received that are longer than the specified length are truncated to the specified length. The number of records truncated is displayed or logged as a part of the reception complete message.

The receive print record length specified should take into consideration:

- Output device capabilities of the local station.
- Transmission record lengths of the remote station.

The RPRRL specified should be equal to or greater than the record lengths being sent to avoid truncation. For transmission/reception of fixed length records, both terminals should agree on the record length.

Paragraph 2.2.16 and table 2-4 provide further information on record lengths.

Examples:

General format:

COMMAND: *RPRRL*
LENGTH: *132*

Condensed format:

COMMAND: *RPRRL*132**

4.5.1.6 Receive Printer Default Pathname Command (RPRD). Use the RPRD command to change the receive printer default pathname. This pathname receives printer data for which no pathname or destination device is specified.

General format:

COMMAND: *RPRD*
PATHNAME: *pathname*

Condensed format:

COMMAND: *RPRD*pathname**

Notes:

Pathnames of the form VOL.DIR.DIR1.FILE$ ($-option) are not allowed as the default pathname. If VOL.DIR.DIR1.FILE$ were entered for the default pathname, the literal FILE$ would be used as the file name instead of assigning consecutive file names of FILE00 through FILEnn. Refer to paragraph 4.3.1 for further information.

The default pathname may not be removed from the receive print queue.

Examples:

General format:

COMMAND: *RPRD*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *RPRD*.DIR.FILE**

4.5.1.7 Receive Punch Default Pathname Command (RPUD). Use the RPUD command to change the receive punch default pathname.

General format:

COMMAND: *RPUD*
PATHNAME: *pathname*

Condensed format:

COMMAND: *RPUD*pathname**

Notes:

Pathnames of the form VOL.DIR.DIR1.FILE$ ($-option) are not allowed as the default pathname. If VOL.DIR.DIR1.FILE$ were entered for the default pathname, the literal FILE$ would be used as the file name instead of assigning consecutive file names of FILE00 through FILEnn. Refer to paragraph 4.3.1 for further explanation.

The default pathname may not be removed from the receive punch queue.

Examples:

General format:

COMMAND: *RPUD*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *RPUD*.DIR.FILE**

4.5.1.8 Receive Punch Record Length Command (RPURL). Use the RPURL command to specify the record length for received card punch data.

General format:

COMMAND: *RPURL*
LENGTH: *record length*

Condensed format:

COMMAND: *RPURL*record length**

Notes:

The record length may be any decimal or hexadecimal number in the range from one through the maximum output buffer size specified in the user parameter table. A hexadecimal number must be preceded by the greater-than sign (>).

The receive punch record length specified should take into consideration:

- Output device capabilities of the local station.
- Transmission record lengths of the remote station.

The RPURL specified should be equal to or greater than the record lengths being sent to avoid truncation. For transmission/reception of fixed length records, both terminals should agree on the record length. Records received that are longer than the specified length will be truncated to the specified length. The number of records truncated during the reception is displayed as part of the reception complete message.

Paragraph 2.2.16 and table 2-4 provide further information about record lengths.

Examples:

General format:

COMMAND: *RPURL*
OPTION: *60*

Condensed format:

COMMAND: *RPURL*60**

4.5.1.9 Records Per Block Command (RB) (2780 Emulator Users Only). Use the RB command to specify the number of records per block for transmitted data.

General format:

COMMAND: *RB*
NUMBER: *number*

Condensed format:

COMMAND: *RB*number**

Notes:

When communicating with an IBM host, a maximum of seven records per block are allowed for transmitted data. Some IBM host systems allow only two records per block.

When communicating with an IBM 2780, two records per block are allowed for transmitted data unless the multiple record transmission feature is installed on the IBM 2780, which then allows up to seven records or 400 characters per block.

Examples:

General format:

COMMAND: *RB*
NUMBER: *5*

Condensed format:

COMMAND: *RB*5**

4.5.1.10 Send Record Length Command (SRL). Use the SRL command to specify a record length value for transmitted data.

General format:

COMMAND: *SRL*
LENGTH: *record length*

Condensed format:

COMMAND: *SRL*record length**

Notes:

The record length may be any decimal or hexadecimal number in the range from one to the maximum input buffer size specified in the user parameter table. A hexadecimal number must be preceded by a greater-than sign (>).

The send record length specified should take into consideration:

- Input record size of the local station.
- Output device capabilities of the remote station.

The SRL specified should be equal to or less than the remote terminal's specified receive record lengths. For fixed length records, both terminals should agree on the record length.

Records transmitted that are longer than the specified length are truncated to the specified length. The number of records truncated during a transmission is displayed as part of the transmission complete message.

Paragraph 2.2.16 and table 2-4 provide further information on record length.

Examples:

General format:

COMMAND: *SRL*
LENGTH: *80*

Condensed format:

COMMAND: *SRL*80**

4.5.1.11 Send Request Command (SRQ). Use the SRQ command to enable or disable the Send Request feature. This feature enables a remote station to transmit a message to the local station that requests that the local station send a specified file to the remote station. The Send Request is identified by a four-character recognition code. Enabling the SRQ command enables the station receiving the Send Request to recognize the four-character recognition code and act on the Send

Request. When the SRQ command is disabled, the four-character recognition code is treated as data. Refer to paragraph 4.5.3.3 for further information.

General format:

COMMAND: *SRQ*

OPTION: { *ON* | *OFF* | *?* }

Condensed format:

COMMAND: *SRQ** { *ON* | *OFF* | *?* } ***

Examples:

General format:

COMMAND: *SRQ*
OPTION: *ON*

COMMAND: *SRQ*
OPTION: *OFF*

Condensed format:

COMMAND: *SRQ*ON**

COMMAND: *SRQ*OFF**

4.5.1.12 Transparency Command (TY). Use the Transparency command to enable or disable transparency for transmitted data. Enabling transparency enables the Emulator to transmit the entire ASCII character set as data.

General format:

COMMAND: *TY*

OPTION: { *ON* | *OFF* | *?* }

Condensed format:

COMMAND: *TY** { *ON* | *OFF* | *?* } ***

Notes:

With transparency off, only a restricted set of ASCII characters (translated to EBCDIC) may be transmitted. With transparency on, the full set of ASCII characters, including control characters (translated to EBCDIC) may be transmitted. Refer to Appendix G for a list of the characters transmitted with transparency off.

Transparency on is not sufficient for transmission of binary data. Refer to paragraphs 4.5.3 and 4.5.4 for information on transmitting/receiving binary data.

Selection of transparency/nontransparency affects only transmitted data; the Emulator operates properly for any state of received data.

When EBCDIC transparency is enabled for the Emulator, any specification of space compression/expansion or end of media will be ignored (i.e., compression/expansion and end of media only apply to nontransparent data).

Transparent data being transmitted or received may be directed to the printer, but any EBCDIC characters less than >40 without an ASCII equivalent are not printed; any EBCDIC characters >40 and greater without an ASCII equivalent are printed as blanks.

Transparent data (not binary) must be transmitted in fixed length records. In 990 to IBM 2780, 3780, or host system communications, the send record length should be determined by the receiving station's output device record lengths; the receive record length (print or punch) is determined by the remote station's input device record lengths. In 990 to 990 communication, the record length is determined by the send record length of the transmitting station; the receive record length should be set accordingly. Refer to table 2-4 for further information on transmit and receive record lengths.

Examples:

General format:

COMMAND: *TY*
OPTION: *ON*

COMMAND: *TY*
OPTION: *OFF*

Condensed format:

COMMAND: *TY•ON•*

COMMAND: *TY•OFF•*

4.5.1.13 Vertical Format Control Command (VFC). Use the VFC command to enable or disable the Vertical Format Control feature. This feature allows the Emulator to retain the vertical format control characters in a received print file which is output to disk. Refer to table 2-2 for a list of the supported vertical format control characters.

General format:

COMMAND: *VFC*

OPTION: { *ON* | *OFF* | *?* }

Condensed format:

COMMAND: *VFC•* { *ON* | *OFF* | *?* } *•*

Examples:

General format:

COMMAND: *VFC*
OPTION: *ON*

COMMAND: *VFC*
OPTION: *OFF*

Condensed format:

COMMAND: *VFC•ON•*

COMMAND: *VFC•OFF•*

4.5.2 CONNECT COMMANDS. The connect commands initiate or accept a communication link connection.

4.5.2.1 Answer Command (ANS). Use the ANS command to establish connection on a communication link by answering a telephone call (the Emulator is programmed to wait for the telephone to ring). The general and condensed formats are the same.

Examples:

COMMAND: *ANS*

4.5.2.2 Call Command (CALL). Use the CALL command to establish connection on a communication link by placing a telephone call.

General format:

COMMAND: *CALL*
NUMBER: *telephone number*

Condensed format:

COMMAND: *CALL*telephone number**

Notes:

The NUMBER: prompt is issued only if the Emulator is configured for an Automatic Call Unit (ACU).

A null response (return) to the NUMBER: prompt specifies manual dialing, even though an ACU has been configured in the Emulator, or specifies that the communication link is a leased line where dialing is not required.

The equal sign character (=) in a telephone number indicates a wait for an intermediate dial tone (tandem dialing). There may be multiple intermediate dial tones in a telephone number. Neither Bell modems nor the external ACU allow a wait for an intermediate dial tone.

Imbedded dashes (-) are allowed in a telephone number for operator readability and are ignored by the Emulator.

Imbedded blanks are not allowed in a telephone number.

The maximum length of a telephone number is 80 characters (the maximum length of an operator command input record).

The asterisk (*) and pound sign (#) are not allowed in the telephone number. The functions of the asterisk and pound sign may be accessed by inputting, in place of the asterisk, any ASCII character whose last four bits are 1010 (such as : or Z), and in place of the pound sign, any ASCII character whose last four bits are 1011 (such as +). These characters may only be used with tone dialing and may be incompatible with some modems.

Examples:

The following example shows a CALL command issued on a system equipped with an ACU.

General format:

COMMAND: *CALL*
NUMBER: *258-1077*

Condensed format:

COMMAND: *CALL*258-1077**

The following example shows a CALL command issued on a system equipped with an ACU. The number is dialed manually or the communication line is a leased line that does not require dialing.

General format:

COMMAND: *CALL*
NUMBER: *return*

Condensed format:

COMMAND: *CALL**

The following example shows a CALL command issued on a system without an ACU. No number prompt is displayed.

General format:	**Condensed format:**
COMMAND: *CALL*	COMMAND: *CALL**

4.5.3 SEND DATA COMMANDS. The send data commands are used to send information over the communication link.

The Send Data (S) and Send Data with Device Selection (SS) commands are executed from the transmit queue. The transmit queue is an ordered queue of all send commands in the order of their entry (first-in, first-out). None of the send transactions are given priority.

In the general format, the send commands are terminated by a null response (return) to a PATHNAME: prompt. In the condensed format, the send commands are terminated by ending a line (record) with an asterisk (*).

The response to a PATHNAME: prompt may be one or more pathnames or text strings separated by commas or asterisks. At least one pathname or text string must be specified. The maximum number of pathnames or text strings allowed with each send command is limited only by the number of queue buffers available at the time the command is entered. When multiple pathnames are specified, the files are concatenated and sent as one file. Any valid pathname may be used in response to a PATHNAME: prompt.

A text string is a valid response to the PATHNAME: prompt for the send commands. A text string must be enclosed in angle brackets (< >) and may not exceed 48 characters. Any ASCII character is allowed in a text string. If using the ampersand (&) to abort the command, the text string must first be closed with an angle bracket and then followed by the ampersand (&).

When a send command is executed, the Emulator parameters (transparency on/off, etc.) are queued with the command. The parameters only affect the transmitted data.

Binary Data Transmissions. In a 990 to 990 communication environment, the Emulator can transmit binary data using the send data commands. Binary is specified using a # (pound sign). The pound sign precedes each pathname being transmitted.

The binary specification is on a per-pathname basis, i.e., each binary file pathname or text string must be preceded by a pound sign (#). The pound sign (#) specifies to the Emulator that the file should not be translated. If pathnames or text strings are being concatenated for transmission (i.e., multiple pathnames or text strings are entered in response to the PATHNAME: prompt or prompts), the first pathname or text string determines whether the transmission will be binary. If the concatenated files are to be transmitted in binary, the first pathname must be preceded with a pound sign (#).

Binary files must be transmitted in fixed length records or single record blocks with transparency on. Transparency does not have to be specified, the Emulator enables it automatically.

The DX10 program files must be transferred as binary files. The transmitting and receiving stations must specify a record length of 288 bytes. To prepare the program file for the transfer, execute a Backup Directory (BD) specifying the complete pathname of the program file to be transferred. The Backup Directory creates a sequential file that can be transmitted. To transmit the file, precede the pathname with a pound sign (#) (specifying do not translate) and use this pathname in the S or SS command. The receiving DX10 must do a Restore Directory (RD) before using the file.

4.5.3.1 Send Data Command (S). Use the S command to transmit a data file or concatenation of files.

General format:

COMMAND: *S*

PATHNAME: {*pathname* / *<text string>*}

.

.

PATHNAME: *return*

Condensed format:

COMMAND: *S*{*pathname* / *<text string>*}*...*...*

Notes:

Data transmitted with the S command has no destination device specification unless the receive pathname feature is used. If the remote terminal is an IBM 3780 or 2780 or another Emulator, the data is routed, by default, to the remote line printer or to a destination specified by the remote line printer queue (if the remote terminal is an Emulator). To specify the destination device, use the SS command.

Examples:

Send the file of pathname .DIR.FILE.

General format:

COMMAND: *S*
PATHNAME: *.DIR.FILE*
PATHNAME: *return*

Condensed format:

COMMAND: *S*.DIR.FILE**

Send the concatenated file .DIR.FILE and text string <THIS IS TEXT>.

General format:

COMMAND: *S*
PATHNAME: *.DIR.FILE*
PATHNAME: *<THIS IS TEXT>*
PATHNAME: *return*

Condensed format:

COMMAND: *S*.DIR.FILE*<THIS IS TEXT>**

Send the concatenated file and text string .DIR.FILE, <THIS IS TEXT>.

General format:

COMMAND: *S*
PATHNAME: *.DIR.FILE,<THIS IS TEXT>*
PATHNAME: *return*

Condensed format:

COMMAND: *S*.DIR.FILE,<THIS IS TEXT>**

Send the file .DIR.FILE as binary data. Transparency will be on.

General format:

COMMAND: *S*
PATHNAME: *#.DIR.FILE*
PATHNAME: *return*

Condensed format:

COMMAND: *S*#.DIR.FILE**

Send the concatenated files .DIR.FILE and .DIR.FILE1. The files will be sent as binary data with transparency on.

General format:

COMMAND: *S*
PATHNAME: *#.DIR.FILE..DIR.FILE1*
PATHNAME: *return*

Condensed format:

COMMAND: *S*#.DIR.FILE..DIR.FILE1**

Send the file .DIR.FILE1 with the destination specified as pathname .DIR.FILE.

General format:

COMMAND: *S*
PATHNAME: *<$$$$.DIR.FILE>*
PATHNAME: *.DIR.FILE1*
PATHNAME: *return*

Condensed format:

COMMAND: *S*<$$$$.DIR.FILE>*.DIR.FILE1**

Send the request that the file .DIR.FILE be queued to be sent to this terminal. The file being queued will have transparency on, compression/expansion on (3780 Emulator users only) or end of media on (2780 Emulator users only), a record length of 132, and the printer selected as the destination device.

General format:

COMMAND: *S*
PATHNAME: *<$$$$$$*T*E*132*PR*.DIR.FILE*>*
PATHNAME: *return*

Condensed format:

COMMAND: *S*<$$$$$$*T*E*132*PR*.DIR.FILE*>**

4.5.3.2 Send Data with Device Selection Command (SS). Use the SS command to cause a data file or concatenation of files to be entered on a queue to be transmitted via the communication device. The destination of the transmitted data is specified by the device selection feature.

General format:

COMMAND: *SS*

DEVICE: $\left\{\begin{matrix} PR \\ PU \end{matrix}\right\}$

PATHNAME: $\left\{\begin{matrix} pathname \\ <text\ string> \end{matrix}\right\}$

.
.
.

PATHNAME: *return*

Condensed format:

COMMAND: $SS*\left\{\begin{matrix} PR \\ PU \end{matrix}\right\}*\left\{\begin{matrix} pathname \\ <text\ string> \end{matrix}\right\}*...*...*$

Notes:

The response to the device prompt may be PR, to specify line printer destination, or PU, to specify card punch destination. These options are translated into the device selection codes when the data is transmitted. Device selection has meaning only when sending data to an IBM 3780 or 2780 or to another Emulator. IBM RJE host systems do not recognize device selection for received data.

Examples:

Send the file .DIR.FILE with the printer selected as the destination device.

General format:

COMMAND: *SS*
DEVICE: *PR*
PATHNAME: *.DIR.FILE*
PATHNAME: *return*

Condensed format:

COMMAND: *SS*PR*.DIR.FILE**

Send the concatenated file .DIR.FILE and text string <THIS IS TEXT> with the card punch selected as the destination device.

General format:

COMMAND: *SS*
DEVICE: *PU*
PATHNAME: *.DIR.FILE*
PATHNAME: *<THIS IS TEXT>*
PATHNAME: *return*

Condensed format:

COMMAND: *SS*PU*.DIR.FILE*<THIS IS TEXT>**

Send the concatenated file and text string .DIR.FILE, <THIS IS TEXT> with the printer selected as the destination device.

General format:

COMMAND: *SS*
DEVICE: *PR*
PATHNAME: *.DIR.FILE,<THIS IS TEXT>*
PATHNAME: *return*

Condensed format:

COMMAND: *SS*PR*.DIR.FILE,<THIS IS TEXT>**

Send the concatenated files .DIR.FILE and .DIR.FILE1 with the printer selected as the destination device. Both files will be sent as binary files since .DIR.FILE is a binary file. Both files will be sent with transparency on.

General format:

COMMAND: *SS*
DEVICE: *PR*
PATHNAME: *#.DIR.FILE,.DIR.FILE1*
PATHNAME: *return*

Condensed format:

COMMAND: *SS*PR*#.DIR.FILE,.DIR.FILE1**

4.5.3.3 Using the Send Request Feature. In a 990 to 990 communication environment, the local station can send a request to the remote station that a specified file be sent to the local station. In essence, the local station sends a Send Data (S) or Send Data with Device Selection (SS) command to the remote station for execution. The send command used in the send request is not a standard Emulator send command. In addition to the standard command, the Send Request send command includes a specification of transparency on or off, end of media (2780) or compression/expansion (3780) on or off, and the send record length.

To use this feature, the operator enters the following in response to the first PATHNAME: prompt of the Send Data (S) command:

- A text string consisting of the Send Request information.
- A pathname of a file whose first record consists of the Send Request information.
- A pathname of a file whose only record consists of the Send Request information.

The Send Request information is as follows:

- A four-character Send Request recognition code. This code may be the standard default recognition code $$$$ or a user-defined recognition code.
- An S or SS.
- An alphabetic character enabling or disabling EBCDIC transparency. Entering the character T enables transparency; the character O disables transparency.
- An alphabetic character enabling or disabling end of media (2780 Emulator users only) or compression/expansion (3780 Emulator users only). Entering the character E enables end of media or compression/expansion; the character O disables end of media or compression/expansion.
- The send record length for the requested transmission.
- The output device selection if SS is the Send Request command. Entering PR selects the printer; PU selects the card punch.
- The pathname or synonym of the file to be transmitted.

The format of the Send Request is as follows:

Using the S command:

RRRRS*T*E*RL*pathname*

Using the SS command:

RRRRSS*T*E*RL*DD*pathname*

where the RRRR field is the four-character Send Request recognition code, S or SS is the send command, the T field is the one-character transparency designation, the E field is the one-character end of media or compression/expansion designation, the RL field is the one- to three-digit send record length, the DD field is the two-character device selection, and pathname is the pathname field.

The Send Request must not contain any blanks. All of the fields shown in the format must be included and the fields must be separated with asterisks as in the preceding format. The terminating asterisk is not required.

The transparency and end of media (2780), or compression/expansion (3780) designations in the Send Request, do not need to consider the condition of these options at the receiving terminal. The Send Request options have no effect on the options at the receiving terminal. The Send Request transaction is transparent to the receiving station except for the logging of messages associated with the transaction and the presence of the S or SS command in the station's transmit queue. The record length specified in the Send Request should consider the maximum input/output buffer size of the receiving terminal to avoid truncation of records.

When a synonym is used as the Send Request pathname, the synonym must be defined at the receiving station.

The receiving station will not recognize the Send Request recognition code unless the Send Request feature is enabled at the receiving station. The Send Request feature is enabled using the Send Request (SRQ) command described in paragraph 4.3.1.11, or it may be enabled as the default value during installation. If the feature is not enabled and the station receives a Send Request, the station treats it as data.

4.5.4 RECEIVE COMMANDS. The receive commands specify the pathnames for received data designated for the printer or punch.

The receive commands are entered on a receive queue as they are executed. The receive queue uses the following priority format:

1st normal pathname entry
.
.

(n)th normal pathname entry
1st $-option entry
.
.
.

(n)th $-option entry
default entry

Each new entry for the receive queue is placed on the queue following all entries of equal or higher priority (lower numerical value) and ahead of all entries of lower priority. For example, a normal pathname entry would be placed on the queue following any entries of the normal variety already on the queue but ahead of any entries of the $-option variety. A $-option entry would be placed on the queue following any normal entries and any $-option entries but ahead of the default entry.

Entries on the queue are accessed in the order of their appearance on the queue. The Emulator looks at the top of the queue to get the next destination for received data. Normal entries are dequeued as they are accessed; $-option entries are dequeued when the $ value exceeds 99 or when an error occurs during an attempt to open the specified pathname. The default entry is never dequeued.

When an open-pathname error is encountered, output proceeds to the next lower priority entry (higher numerical value) on the receive print queue. If there are no more entries on the queue, an error is reported to the host system.

The pathname specification may be of the form: VOL.DIR.DIR1.FILE$. If this pathname form is specified, received data is routed to consecutive files, beginning with VOL.DIR.DIR1.FILE00 and ending with VOL.DIR.DIR1.FILE99 (or some value less than 99 if the capacity of the device is exceeded), and then continues to the next entry on the queue. The maximum length of FILE in pathnames of this form is six characters.

Binary Data Receptions. In a 990 to 990 communication environment, the Emulator can receive binary data. Binary is specified using a # (pound sign). The pound sign precedes the receive pathname specified by the RPR or RPU command. The pound sign (#) specifies to the Emulator that the file should not be translated.

Binary data is automatically transmitted with transparency on. Binary data must be transmitted in fixed length records or single record blocks. The transmitting and receiving stations should agree on a record length.

Received binary files should not be directed to the printer unless binary or nontranslated data is expected. Since these files are not being translated, printing results may be unpredictable.

The DX10 program files must be transferred as binary files. The transmitting and receiving stations must specify a record length of 288 bytes. To prepare the program file for the transfer, execute a Backup Directory (BD) specifying the complete pathname of the program file to be transferred. The Backup Directory creates a sequential file that can be transmitted. To transmit the file, precede the pathname with a pound sign (#) (specifying do not translate) and use this pathname in the S or SS command. The receiving DX10 must do a Restore Directory (RD) before using the file.

4.5.4.1 Receive Printer Data Command (RPR). Use RPR command to specify the destination pathname (device or file) for each received data file with line printer device selection.

General format:

COMMAND: *RPR*
PATHNAME: *pathname*

Condensed format:

COMMAND: *RPR*pathname**

Notes:

The vertical format control characters are deleted when the destination is a data file, magnetic tape, or cassette (unless VFC=ON is specified). When the destination is a VDT or TTY terminal, the output is single spaced. The vertical format control characters are translated to software forms control formatting for files directed to the line printer.

Examples:

Receive print file in pathname .DIR.FILE.

General format:

COMMAND: *RPR*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *RPR*.DIR.FILE**

Receive binary print file in pathname .DIR.FILE.

General format:

COMMAND: *RPR*
PATHNAME: *#.DIR.FILE*

Condensed format:

COMMAND: *RPR*#.DIR.FILE**

Receive consecutive print files in pathnames .DIR.FILE00 through .DIR.FILEnn.

General format:

COMMAND: *RPR*
PATHNAME: *.DIR.FILE$*

Condensed format:

COMMAND: *RPR*.DIR.FILE$**

4.5.4.2 Receive Punch Data Command (RPU). Use the RPU command to specify the destination pathname (device or file) for each received data file with card punch device selection.

General format:

COMMAND: *RPU*
PATHNAME: *pathname*

Condensed format:

COMMAND: *RPU*pathname**

Examples:

Receive punch file in pathname .DIR.FILE.

General format:

COMMAND: *RPU*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *RPU*.DIR.FILE**

Receive binary punch file in pathname .DIR.FILE.

General format:

COMMAND: *RPU*
PATHNAME: *#.DIR.FILE*

Condensed format:

COMMAND: *RPU*#.DIR.FILE**

Receive consecutive punch files in pathnames .DIR.FILE00 through .DIR.FILEnn.

General format:

COMMAND: *RPU*
PATHNAME: *.DIR.FILE$*

Condensed format:

COMMAND: *RPU*.DIR.FILE$**

4.5.4.3 Using the Receive Pathname Feature. The Receive Pathname feature enables the transmitting station to specify the receive pathname for a transmitted file.

To use this feature, the operator enters a four-character recognition code followed by a destination pathname as the response to the first PATHNAME: prompt of a Send Data (S) command. This response may be one of the following:

- A pathname or synonym of a file whose first record consists of the Receive Pathname recognition code followed by the destination pathname.

- A pathname or synonym of a file whose only record consists of the Receive Pathname recognition code followed by the destination pathname.
- A text string consisting of the Receive Pathname recognition code followed by the destination pathname.

The format of the Receive Pathname is:

nnnn { pathname / synonym }

where nnnn is the recognition code. This recognition code is set during Emulator installation.

The receiving station will not recognize the Receive Pathname four-character recognition code unless the Receive Pathname feature is enabled at the receiving station. The Receive Pathname feature is enabled using the Receive Pathname (RPN) command described in paragraph 4.3.1.4 or it may be enabled as the default value during installation. If the feature is not enabled and the station receives a Receive Pathname designation, the station will treat the designation as data.

With the Receive Pathname enabled, the transmitted file is placed in the specified pathname without affecting the receiving station's receive queue. If a file is not transmitted in the same Send Data (S) command which transmitted the Receive Pathname designation, the next file transmitted will be placed in the Receive Pathname specified.

When a synonym is used as the destination pathname, it must be defined at the receiving station.

4.5.5 COMMAND FILE COMMANDS. Command file commands are used to specify a command file as the secondary command source, to open a log file, and to aid in the execution of a command file.

4.5.5.1 Command File Command (CF). Use the CF command to specify a pathname (device or file) as the command source (secondary). Refer to paragraph 4.4.2 for information on using command files.

General format:

COMMAND: *CF*
PATHNAME: *pathname*

Condensed format:

COMMAND: *CF*pathname**

Notes:

Command file commands cannot be nested. If a command file, specified by a CF command, issues a CF command, the active command file closes and the new command file begins issuing commands. Control will not be returned to the original command file.

A null (blank) pathname response to the CF command causes the active command file to be closed and session control to return to the primary command source.

Examples:

General format:

COMMAND: *CF*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *CF*.DIR.FILE*

4.5.5.2 Conditional Command (IF). Use the IF command to conditionally execute a command or group of commands. If the specified condition is true, the Emulator executes all commands until $ELSE is encountered in the command stream and then skips all commands until $END is encountered. If the specified condition is false, the Emulator skips all commands until $ELSE or $END is encountered.

General format:

COMMAND: *IF*
OPTION: { *condition* / *?* }

Condensed format:

COMMAND: *IF*{ *condition* / *?* }*

Options (conditions):

CON Condition is true if the communication link is connected.

DISC Condition is true if the communication link is not connected.

S Condition is true if the transmit queue is empty.

RPR Condition is true if the receive print queue is empty. Since the default entry may never be removed from the queue and entries of the form pathname$ are removed only under certain conditions, receive queues are considered empty when all entries of normal pathname form are completed.

RPU Condition is true if the receive punch queue is empty (refer to the preceding RPR option for the definition of empty).

ALLQ Condition is true if all queues are empty (refer to the RPR option for the definition of empty for receive queues).

Notes:

Nested IF commands are not allowed. If a second IF command is encountered in an IF—$ELSE—$END sequence, it is treated as an invalid command input and an appropriate message is logged.

If an IF command is issued and the condition is false, the Emulator will execute no commands until an $ELSE or an $END is executed.

The $ELSE and $END are entered in response to the COMMAND: prompt as commands.

Examples:

General format:

COMMAND: *IF*
OPTION: *CON*

COMMAND: *IF*
OPTION: *DISC*

Condensed format:

COMMAND: *IF*CON**

COMMAND: *IF*DISC**

The following example shows how the commands would appear in a command file (the condensed format is recommended for command files):

Condensed format:

```
IF*CON
S*.DIR.FILE*
$ELSE
ST*ALL
$END
```

4.5.5.3 Log File Command (LF). Use the LF command to specify that all operator message output be directed to a secondary log pathname (device or file) while operating from a command file or intertask message queue.

General format:

COMMAND: *LF*
PATHNAME: *pathname*

Condensed format:

COMMAND: *LF*pathname**

Notes:

If the LF command is entered while in Operator mode, the log file is opened but log output is not routed to the log file until entering Command File mode or Intertask Message Queue mode.

If the LF command is entered while a secondary log file is already active, the old log file is closed and the new log file is opened.

If the LF command is entered with a null response (return) to the PATHNAME: prompt (LF* in condensed form), any currently open secondary log file is closed and log output reverts to the primary log destination.

If an I/O error is encountered in an attempt to write to a secondary log file, the log file is closed and log output reverts to the primary log destination.

Examples:

General format:

COMMAND: *LF*
PATHNAME: *.DIR.FILE*

Condensed format:

COMMAND: *LF*.DIR.FILE**

4.5.5.4 Wait Command (WAIT). Use the WAIT command to suspend all command activity until a specified condition is met. The WAIT command allows delaying operation until the communication line is connected or disconnected or until a specific Emulator queue or all queues are empty. The WAIT command is intended for use in command files when it is necessary to wait for a specific condition before issuing a command that may be dependent on that condition.

General format:

COMMAND: *WAIT*
OPTION: {*condition* / *?*}

Condensed format:

COMMAND: *WAIT*{*condition* / *?*}*

Options (conditions):

CON — Wait until connect complete. The WAIT command with the CON option should be issued only after issuing a command that accomplishes connection.

DISC — Wait for communication link disconnection.

S — Wait for transmit queue empty.

RPR — Wait for receive print queue empty. Since the default entry may never be removed from the queue and entries of the form pathname$ are removed only under certain conditions, receive queues are considered empty when all entries of normal pathname form are completed.

RPU — Wait for receive punch queue empty (see the preceding RPR option for the definition of empty).

ALLQ — Wait for all queues to be empty (see the RPR option for the definition of empty for receive queues).

Notes:

The Emulator waits for connect only if a condition exists that will accomplish connection (connect in progress).

The Emulator does not wait for queue empty (S, RPR, RPU, ALLQ) if a condition exists such that the queue can never be emptied, such as communications disconnected. The Emulator waits only if connected or a connect is in progress.

Examples:

General format:

COMMAND: *WAIT*
OPTION: *CON*

COMMAND: *WAIT*
OPTION: *RPR*

Condensed format:

COMMAND: *WAIT*CON**

COMMAND: *WAIT*RPR**

4.5.6 DISCONNECT COMMANDS. The disconnect commands accomplish a disconnect when issued.

4.5.6.1 Abort Command (ABORT). Use the ABORT command to cause an immediate disconnection, an immediate termination of the Emulator task, or a conditional dequeuing of a specific transaction, depending on the option specified.

General format:

COMMAND: *ABORT*
OPTION: {*option* / *?*}

Condensed format:

COMMAND: *ABORT*{*option* / *?*}*

Options:

COM — Causes immediate disconnection of the communication link. ABORT COM does not wait for task queues to become empty or for any transmit or receive in progress to complete. No transactions are removed from the task queues. The Emulator task remains active and accepts further command input.

TASK — Causes immediate termination of the Emulator task. ABORT TASK does not wait for task queues to become empty or for any transmit or receive in progress to complete. The task queues are deleted.

ID — Causes the transaction ID specified to be removed from the task queues. If the specified transaction is currently being transmitted or received, or if the receive print or receive punch default pathname is specified, the ABORT transaction ID is not performed and a message is logged.

Notes:

ABORT transaction ID does not abort a transaction that is in progress but does remove the transaction from the queue if it is not in progress. To abort a transaction that is in progress, an operator would first issue the ABORT COM command (the transaction would then no longer be in progress) and then issue the ABORT transaction ID command.

Examples:

General format:

COMMAND: *ABORT*
OPTION: *COM*

Condensed format:

COMMAND: *ABORT*COM**

General format:

COMMAND: *ABORT*
OPTION: *10*

Condensed format:

COMMAND: *ABORT*10**

4.5.6.2 Disconnect Command (DISC). Use the DISC command to disconnect a communication link (hang up the telephone). When the DISC command is entered, the Emulator waits for all enqueued transmit transactions to be completed, waits for any receive transactions in progress to be completed, and then disconnects the communications link. The general and condensed formats are the same.

Format:

COMMAND: *DISC*

4.5.6.3 Terminate Task Command (TERM). Use the TERM command to terminate the Emulator task. When the TERM command is entered, the Emulator waits for all enqueued transmit transactions to be completed, waits for any receive transactions in progress to be completed, and then disconnects the communication link and terminates the Emulator task. The general and condensed formats are the same.

Format:

COMMAND: *TERM*

4.5.7 MISCELLANEOUS COMMANDS. The Reset command and the Statistics command perform special functions for the Emulator as described in the following paragraphs.

4.5.7.1 Reset Command (RESET). Use the RESET command to cause an EOT to be transmitted via the communication link at the next possible opportunity. The EOT causes the communication link to be reset. The general and condensed formats are the same.

Format:

COMMAND: *RESET*

4.5.7.2 Statistics Command (ST). Use the ST command to cause the entire set or a selected subset of the communication statistics to be logged.

General format:

COMMAND: *ST*
OPTION: {*option* / *?*}

Condensed format:

COMMAND: *ST** {*option* / *?*} *

Options:

CUR	Displays/logs the current statistics.
CT	Displays/logs the communication statistics since the last communication link connection.
S	Displays/logs a list of the transactions in the transmit or send queue.
RPR	Displays/logs a list of the transactions on the receive printer queue.
RPU	Displays/logs a list of the transactions on the receive card punch queue.
ALLQ	Displays/logs lists of the transactions on all of the queues.
ALL	Displays/logs a comprehensive set of statistics (all of the above).

Note:

A null response to the option prompt specifies the ALL option.

The following are examples of three of the options. The printout are of the statistics that could be displayed and are followed by an explanation of some of the statistics shown.

Example:

General format:

COMMAND: *ST*
OPTION: *CUR*

Condensed format:

COMMAND: *ST*CUR**

Statistics Display at the DX10 2780 Emulator

```
DX10/2780        2.0.0  79.046

COMM DISCONNECTED
TRANSPARENCY OFF
END OF MEDIA OFF
RECEIVE PATHNAME OFF
SEND REQUEST OFF
PRINT FILE VFC OFF
BLOCK LENGTH OFF 00400
TRANSMIT RECORD LENGTH 00080
RECEIVE PRINT RECORD LENGTH 00132
RECEIVE PUNCH RECORD LENGTH 00080
RECORDS PER BLOCK 00007
```

Statistics Display at the DX10 3780 Emulator

```
DX10/3780        2.0.0  79.046

COMM DISCONNECTED
TRANSPARENCY OFF
COMPRESSION/EXPANSION OFF
RECEIVE PATHNAME OFF
SEND REQUEST OFF
PRINT FILE VFC OFF
BLOCK LENGTH 00512
TRANSMIT RECORD LENGTH 00080
RECEIVE PRINT RECORD LENGTH 00132
RECEIVE PUNCH RECORD LENGTH 00080
```

The information displayed represents the status of the communication link and those values either set by a command or set during installation that have not been varied by a command during this session. In addition to the statistics shown in the example, the following information may also be displayed in response to the CUR option:

- Pathname of any currently open command file.
- Pathname of any currently open log file.
- Transmit pathname if transmit active or transmit pending.
- Receive print pathname if active.
- Receive punch pathname if active.

Example:

General format:

COMMAND: *ST*
OPTION: *CT*

Condensed format:

COMMAND: *ST*CT**

Statistics Display

```
    COMM LINK STATISTICS
    RECEIVE TRANSMIT
1.  00000   00000   BID
2.  00000   00000   BID ACCEPTED
3.  00000   00000   TTD
4.  00000   00000   NAK
5.  00000   00000   WACK
6.  00000   00000   ENQ
7.  00000   00000   EOT
8.  00000   00000   ACK
9.  00000           CONVERSATIONAL REPLY
10. 00000           RVI
11. 00000           BID OVERRIDE
12. 00000           BID THRESHOLD EXCEEDED
13. 00000           NAK THRESHOLD EXCEEDED
14. 00000           ENQ THRESHOLD EXCEEDED
15. 00000           IDLE THRESHOLD EXCEEDED
```

These statistics reference Binary Synchronous Communications procedures. (For an explanation of these procedures, refer to Appendix F.) These statistics reflect the following information as indicated by the corresponding numbers.

1. *Number of BIDs (received/transmitted).* A count of the ENQs received and sent as a bid for the use of the communication line.

2. *Number of BIDs accepted (received/transmitted).* A count of the positive responses received and transmitted in response to a bid sequence.

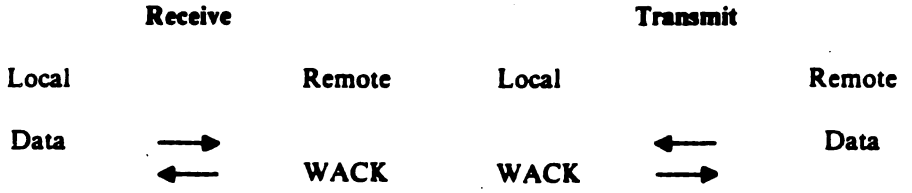
Receive			**Transmit**		
Local		Remote	Local		Remote
BID	→			←	BID
	←	ACK0	ACK0	→	

3. *Number of TTDs (received/transmitted).* A count of the TTD sequences received and transmitted. The TTD sequence is transmitted by the transmitting station to indicate that it is temporarily not ready to transmit. The 2780 Emulator cannot transmit a TTD.

Receive			**Transmit**		
Local		Remote	Local		Remote
	←	Data	Data	→	
ACK0	→			←	ACK0
	←	TTD	TTD	→	
NAK	→			←	NAK

4. *Number of NAKs (received/transmitted).* A count of the negative responses received and transmitted. Since the NAK can be received/transmitted in response to more than one situation, determine the number of NAKs received in response to data sent by subtracting the number of TTDs transmitted from the number of NAKs received. For the number of NAKs transmitted in response to data received, subtract the number of TTDs received from the number of NAKs transmitted.

5. *Number of WACKs (received/transmitted).* A count of the number of WACKs received and transmitted. The WACK is transmitted by the receiving station to indicate that it is temporarily not ready to receive. The 2780 Emulator cannot transmit a WACK.

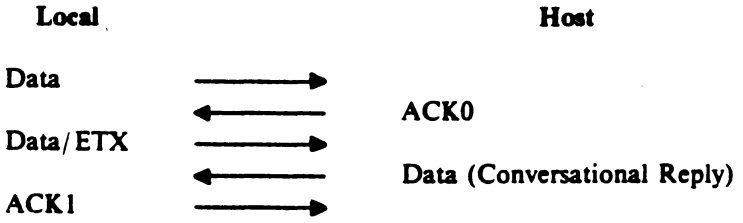


6. *Number of ENQs (received/transmitted).* A count of the ENQs received and transmitted. The ENQ is transmitted by the station to bid for the line in response to a message containing errors or as part of the TTD sequence.

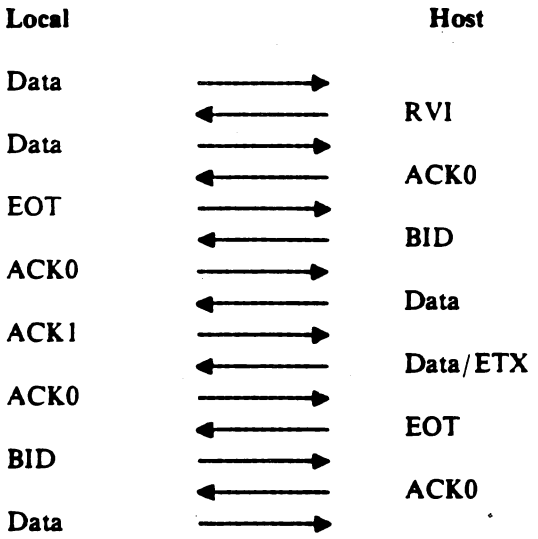
7. *Number of EOTs (received/transmitted).* A count of the EOTs received and transmitted. The EOT is transmitted to terminate a transmission or to indicate error conditions.

8. *Number of ACKs (received/transmitted).* A count of the ACKs received and transmitted. An ACK is transmitted as a positive response to a text block (i.e., the block was received without error) or in response to a line bid.

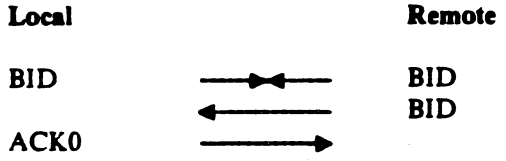
9. *Number of Conversational Replies (received).* A count of the Conversational Replies received. The host can respond with a conversational reply in place of a positive acknowledgement after it receives an ETX (3780 Emulator users only).



10. *Number of RVIs (received).* A count of the RVIs received. The host system interrupts the local station's transmissions to send priority data. When the host system completes its data transfer, the local station again BIDs for the line and resumes transmitting where it was interrupted.



11. *Number of BIDs overridden.* A count of the BIDs that have been overridden. Both stations have BID at the same time. The station designated as the primary station waits one second and reBIDs. This BID overrides the secondary station's BID.



12. *Number of times the BID threshold has been exceeded.** The line is BID for unsuccessfully n times in a row. The value of n is the threshold.

13. *Number of times the NAK threshold has been exceeded.** The station responds with NAK n times in a row. The value of n is the threshold.

14. *Number of times the ENQ threshold has been exceeded.** The station has responded with ENQ n times in a row. The value of n is the threshold.

15. *Number of times the IDLE threshold has been exceeded.** The station has not transmitted anything for n seconds. The value of n is the threshold.

*These thresholds are set during the building of the Emulator as described in Appendix C.

Example:

General format:

COMMAND: *ST*
OPTION: *ALLQ*

Condensed format:

COMMAND: *ST*ALLQ**

```
TRANSMIT QUEUE
EMPTY

RECEIVE PRINT QUEUE
00001 LP01

RECEIVE PUNCH QUEUE
00002 DUMY
```

The ALLQ option gives the statistics on the status of the queues.

The format of the information displayed is as follows:

xxxxx QUEUE
nnnnn pathname . . . pathname

where xxxxx is TRANSMIT, RECEIVE PRINT, or RECEIVE PUNCH; nnnnn is the identification number of the transaction; and the pathname is the pathname or pathnames queued with the transaction. For transactions with multiple pathnames, all pathnames are displayed on the same line. If the pathnames exceed the 80 character log record length, the record is truncated at 80 characters.

The individual queues may be displayed with the S, RPR, and RPU options. The queue information is formatted as it is for the ALLQ option. If the queue specified has no entries, the Emulator displays the message QUEUE EMPTY.

APPENDIX A

EMULATOR HARDWARE REQUIREMENTS

The Emulator runs with the DX10 standard equipment configuration plus communication equipment as follows:

- A model 990 10 computer with:

  64K words of memory (minimum)
  System disk
  System console (VDT or ASR/KSR data terminal)
  Means of disk backup

- One of the following communication interface sets:

  Communication interface module (946104-0001), synchronous modem (945094-0003), and optional auto-call unit (945163-0001).

  Communication interface module and cable for non-TI modems (946104-0002) and optional auto-call unit for non-TI modems (2263480-0001). Tandem dialing is not supported for the auto-call unit used with non-TI modems.

The supported non-TI modems are the Bell data sets listed. Following each modem is a list of the options recommended. An asterisk beside an option indicates that the option is required for use with the Emulator.

- 201A switched-line modem (2000 bps)

  *Internal timing
  Without new sync
  *EIA option
  *Without ACU
  Alternate voice
  Selective unattended answer
  Half-duplex operation
  Carrier controlled by request to send

- 201B leased-line modem (2400 bps)

  *Internal timing
  *Without new sync
  *EIA option

- 201C switched-line or leased-line modem (2400 bps)

  Switched network options:

  *Internal timing (supplied by data set)
  *EIA voltage for CD and RDY

*EIA voltage for ring
*Without ACU
Grounding option AB not connected to AA
Permanent auto-answer

Dedicated private line service:

*Internal timing
*Without new sync
Grounding option AB not connected to AA

- 208A leased-line modem (4800 bps)

  *Internal timing
  *Without new sync
  Grounding option AB not connected to AA
  CC off when AL button is pressed

- 208B switched-line modem (4800 bps)

  *Internal timing
  Grounding option AB not connected to AA
  CC off when AL button is pressed
  Auto-answer under CD control

Cassette tape may be used for Emulator input/output only if the 733 ASR is not also the Emulator operator console.

Input/output using slow devices could cause error messages on the host console.

Refer to Section II, Concepts and Facilities, of the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume I,* part number 946250-9701, for information on DX10 supported hardware.

APPENDIX B

EMULATOR SOFTWARE REQUIREMENTS

The DX10 3780/2780 Emulator software includes a separate task and Device Service Routine (DSR). The task contains the Emulator functions, the operator interface, and the tables and buffers. The code is organized into task (tables, buffers, and overlay areas) and procedure (reentrant code). The operator interface portion of the task makes extensive use of overlays.

The Emulator software is designed to the following specifications:

- The DSR requires no more than 7K (7168) bytes of memory (6KB base code plus 1KB tables, with an additional 1KB tables to be allocated from the DX10 system's table area). Support of multiple communication interfaces requires 6KB shareable base plus 1KB per communication interface.

- The task requires no more than 16K (16384) bytes of memory (8KB base code plus 8KB tables and buffers). Support of multiple communication channels requires 8KB shareable base plus 8KB per communication channel.

- The communication buffers are, by default, 400 bytes for the 2780 and 512 bytes for the 3780. These buffers may be changed to sizes smaller than the default value using the operator command, BL. Larger buffers require editing, assembly, and linking and will increase the memory requirements.

APPENDIX C

EMULATOR INSTALLATION

C.1 INTRODUCTION

The installation of the DX10 3780 and 2780 Emulators is accomplished by:

- Building the DSR
- Generating the DX10 system
- Installing the appropriate task (3780 or 2780).

Batch streams and any necessary link control files are provided with the Emulator Package Release to aid these installation processes. Tasks reflecting the standard configuration and parameter default values for the 2780 and 3780 are also supplied.

C.2 BUILDING THE DEVICE SERVICE ROUTINE (DSR)

In building the DSR, you may use the standard configuration and protocol parameters or modify them according to the requirements of your application. If any of the standard parameters are not acceptable, edit the DSR source parameters to reflect your needs, assemble the DSR parameters module, and link the DSR.

The modifiable parameters and their standard or default values are as follows:

Parameter	Standard Value	User-Defined* Value
Communication Device CRU Address	>40	_____
BID Timeout (primary or secondary)	Secondary/one second	_____
ACK Timeout	Three seconds	_____
Data Timeout (To operate as a host system, it may be desirable to set this value to 30 seconds.)	Infinite	_____
IDLE Line Timeout	20 seconds	_____
BID Retry Count	15	_____
ENQ Retry Count	15	_____
NAK Retry Count (To operate as a host system, it may be desirable to set this value to 15.)	Infinite	_____

*This column is provided to record any user-defined values set during the building of the DSR.

For further information about building the DSR, refer to the *DX10 3780/2780 Emulator Release 2.0.0 Object Installation Manual*, part number 2250918-9701.

C.3 GENERATING THE DX10 SYSTEM

For information on generating the DX10 system, refer to Section I of the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume V System Programming Guide*, part number 946250-9705.

C.4 INSTALLING THE TASK

The tasks for the 3780 and 2780 are supplied with the release. Each task includes a table of system parameters reflecting the standard configuration and parameter default values. If the standard table is not acceptable for your particular application, you can edit the parameter table to reflect your needs. The parameter table must then be assembled and the task relinked.

The parameters, their standard or default values, and their value ranges are as follows:

Parameter	Emulator	Value Range	Standard Value	User-Defined* Value
Disconnect on Error	3780/2780	Zero = do not disconnect Nonzero = disconnect	Zero	_____
Line Connection Timeout	3780/2780	Zero (infinite) through 255 minutes	Zero (infinite)	_____
Punch Selection Character	3780	DC2 or DC3	DC2	_____
Auto-Call Unit (ACU) CRU Address	3780/2780	>FFFF = no ACU available >xxxx = the correct ACU CRU address	>FFFF	_____
ACU Dial Type	3780/2780	Zero = tone dialing Nonzero = pulse dialing	Nonzero	_____
ACU Retry Count	3780/2780	No minimum/maximum	3 Retries	_____
ACU Retry Interval	3780/2780	No minimum/maximum	3 Seconds	_____
Transparency or Nontransparency	3780/2780	Zero = nontransparency Nonzero = transparency	Zero	_____
Space Compression/ Expansion	3780	Zero = disabled Nonzero = enabled	Zero	_____
End of Media	2780	Zero = disabled Nonzero = enabled	Zero	_____
Receive Pathname Feature	3780/2780	Zero = disabled Nonzero = enabled	Zero	_____
Send Request Feature	3780/2780	Zero = disabled Nonzero = enabled	Zero	_____
4-Character Recognition Code for Receive Pathname Feature	3780/2780	Any legal 4-character string	$$$$	_____
4-Character Recognition Code for Send Request Feature	3780/2780	Any legal 4-charcter string	$$$$	_____
Line Printer Data Default Pathname	3780/2780	Any valid pathname	LP01	_____
Punch Data Default Pathname	3780/2780	Any valid pathname	DUMY	_____
Strip Printer Vertical Forms Control Characters from Disk Files	3780/2780	Zero = strip Nonzero = do not strip	Zero	_____

Parameter	Emulator	Value Range	Standard Value	User-Defined* Value
Maximum Communication Block	3780	512 bytes - 1024 bytes	512 bytes	_____
	2780	400 bytes - 1024 bytes	400 bytes	_____
Transmitted/Received Data Maximum Record Lengths	3780	Input: 80 - 508 bytes	300 bytes	_____
		Output: 80 - 508 bytes	300 bytes	_____
	2780	Input: 80 - 396 bytes	300 bytes	_____
		Output: 80 - 396 bytes	300 bytes	_____
Transmitted/Received Data Default Record Lengths	3780/2780	Card reader: 80 - max**	80 bytes	_____
		Line printer: 80 - max**	132 bytes	_____
		Card punch: 80 - max**	80 bytes	_____
Records Per Block Maximum	2780	Minimum = 1 Maximum = 0 (as many as the block will hold)	7 records	_____
Buffer Pool Size	3780/2780	672 bytes - 2000 bytes	1040 bytes	_____

*This column is provided to record user-defined values set during installation.

**The maximum value is determined by the value set for the Transmitted/Received Data Maximum Record Lengths.

For further information about generating the Emulator task, refer to the *DX10 3780/2780 Emulator Object Installation Guide*, part number 2250918-9701.

APPENDIX D

COMMUNICATION SESSION

The following communication session demonstrates a possible application of the Emulator. The session has two DX10 3780 Emulators in communication. System 1 is the local station; system 2 is the remote station.

The session is illustrated on the following four pages. The session begins with sheets 1 and 2. Sheet 1 shows the operations at system 1. Sheet 2 shows the operations at system 2. Special indicators are used to show the interactions between system 1 and system 2 and within system 1. These indicators consist of the following symbols:

Ⓧn — or — Ⓧn

n Ⓧ — or — n Ⓧ

These indicate a system-to-system or page-to-page relationship. The Ⓧn — or — Ⓧn symbol means that the data to the right or left of the arrow results in the data on sheet n pointed to by letter x. On sheet n, a corresponding symbol n Ⓧ — or — n Ⓧ points to the resulting data. The n Ⓧ — or — n Ⓧ symbol means that the data to the right or left of the arrow results from the data on page n pointed to by letter x.

The braced areas, A, B, C, D, and E, indicate the command sources and log destinations. Area A is the primary command source (operator) and primary log destination (operator terminal) for system 1. Area B is the secondary command source (command file .CMDFIL1) for system 1. Area C is the secondary log destination (log file .LOGFIL) for system 1. Area D is the primary command source (operator) and primary log destination (operator terminal) for system 2. Area E is the secondary command source (command file .CMDFIL2) for system 1. (.CMDFIL2 replaces .CMDFIL1 as the secondary command source.)

Sheets 3 and 4 are a continuation of the communication session. Sheet 3 is a continuation of system 1's operations; sheet 4 is a continuation of system 2's operations.

Communication Session: Sheet 1

The operator activates system 1 by entering X3780 in response to the command mode ([ ]) and then by responding to the activation prompts. Once the emulator is executed, the operator specifies a command file, .CMDFIL1. When the command file is opened, system 1 operates in unattended mode while the command file .CMDFIL1 issues the commands. The first command of .CMDFIL1 opens a log file .LOGFIL. The log of the LF command is displayed on the operator terminal because the log file is not yet open and the default log output is the operator terminal.

.CMDFIL1 sets the parameters for system 1's interactions, i.e.:

- System messages will be logged in .LOGFIL.
- Received print files without a specified destination will be received in pathnames .RPRINT00 through .RPRINTnn.
- The receive pathname feature is enabled.
- The compression/expansion feature is enabled.
- A punch file will be received in pathname .CMDFIL2.

.CMDFIL1 then issues commands to answer the incoming call, to send a message indicating its readiness to receive, to wait for the reception of .CMDFIL2 from system 2, to specify .CMDFIL2 as the next command source when it is received, and to terminate.

After system 2 transmits the command file .CMDFIL2, system 1 opens .CMDFIL2 and begins executing its commands. .CMDFIL2 issues three send commands and a statistics command and then terminates.

.LOGFIL is the log of the transactions at system 1 beginning with the first command following the LF command issued by .CMDFIL1 through the disconnection (*** LINE DISCONNECTED) initiated by the TERM command issued by .CMDFIL2.

A. SYSTEM 1 OPERATOR CONSOLE

```
[] X3780
EXECUTE 3780 EMULATOR CURRENT VERSION
          COMMAND ID: ME
              LOG ID: ME
        COMM DEVICE ID: CM01

== FOREGROUND COMMAND EXECUTING ==

 DX10/3780        2.0.0  79.046

 COMMAND:  CF
 PATHNAME: .CMDFIL1

 COMMAND:
 LF*.LOGFIL
```

B. SYSTEM 1 COMMAND FILE .CMDFIL

```
LF*.LOGFIL
RPR*.RPRINT$
RPN*ON
CE*ON
RPU*.CMDFIL2
ANS*
S*'READY FOR DATA'*
WAIT*RPU
CF*.CMDFIL2
```

C. SYSTEM 1 LOG FILE .LOGFIL

```
COMMAND:
RPR*.RPRINT$
ID 00003

COMMAND:
RPN*ON

COMMAND:
CE*ON

COMMAND:
RPU*.CMDFIL2
ID 00004

COMMAND:
ANS*

COMMAND:
S*'READY FOR DATA'*
ID 00005

COMMAND:
WAIT*RPU

*** CONNECT SUCCESSFUL

*** TRANSMIT COMPLETE 00 00005 00001 00000 READY FOR DATA

*** RPNAME COMMAND .DATATST1

*** RPRINT COMPLETE 00 00006 00009 00000 .DATATST1

*** RPNAME COMMAND .DATATST2

*** RPRINT COMPLETE 00 00007 00053 00000 .DATATST2
```

(B)140892(1,4)

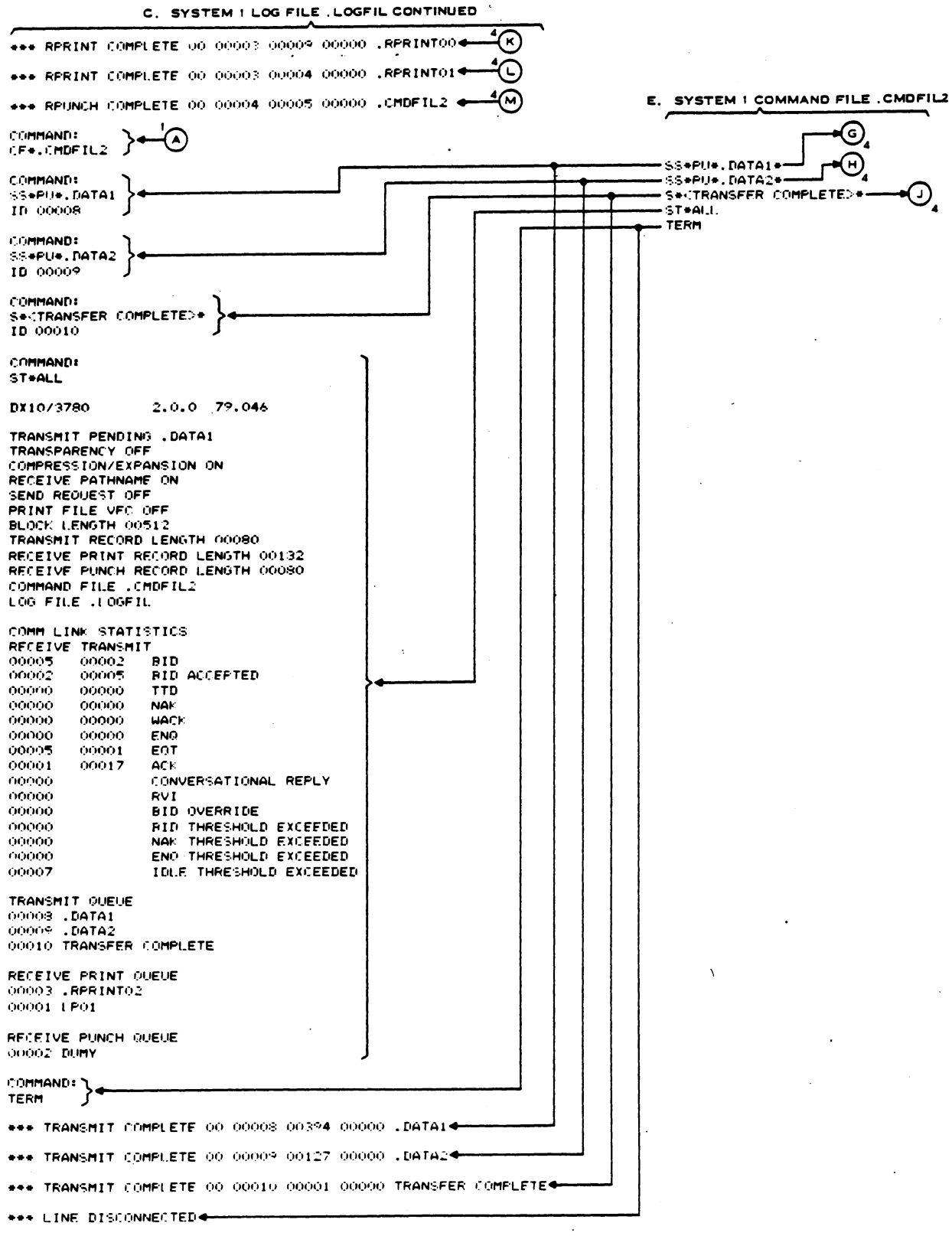
Communication Session: Sheet 2

D. SYSTEM 2 OPERATOR CONSOLE

```
[]X3780

EXECUTE 3780 EMULATOR CURRENT VERSION
            COMMAND ID:  ME
                LOG ID:  ME
        COMM DEVICE ID:  CM01

== FOREGROND COMMAND EXECUTING ==

 DX10/3780      2.0.0  79.046

(D)1 COMMAND:  CALL

     COMMAND:

1
(B)  *** CONNECT SUCCESSFUL

     COMMAND:

1
(C)  *** RPRINT COMPLETE 00 00001 00001 00000 LP01

     COMMAND:  RPU
     PATHNAME: .RPUNCH$
     ID 00003

     COMMAND:  $
     PATHNAME: <$$$$.DATATST1>
     PATHNAME: .COMM.COMMTEST
     PATHNAME:
(E)1 ID 00004

     COMMAND:

     *** TRANSMIT COMPLETE 00 00004 00010 00000 $$$$.DATATST1

     COMMAND:  $
     PATHNAME: <$$$$.DATATST2>
     PATHNAME: .COMM.DATATST2
     PATHNAME:
(F)1 ID 00005

     COMMAND:

     *** TRANSMIT COMPLETE 00 00005 00054 00000 $$$$.DATATST2
```

(B)140892-2 4

Communication Session: Sheet 3



Communication Session: Sheet 4

D. SYSTEM 2 OPERATOR CONSOLE CONTINUED

```
COMMAND:  S
PATHNAME: .COMM.COMMTEST
PATHNAME:
ID 00006                                                    (K) 3

COMMAND:

*** TRANSMIT COMPLETE 00 00006 00009 00000 .COMM.COMMTEST

COMMAND:  S
PATHNAME: <THIS PRINT
PATHNAME: <WILL BE
PATHNAME: <DIRECTED TO
PATHNAME: <.RPRINT01>                                       (L) 3
PATHNAME:
ID 00007

COMMAND:

*** TRANSMIT COMPLETE 00 00007 00004 00000 THIS PRINT

COMMAND:  SS
DEVICE:   PU
PATHNAME: DS02.CMDFIL2
PATHNAME:                                                   (M) 3
ID 00008

COMMAND:

*** TRANSMIT COMPLETE 00 00008 00005 00000 DS02.CMDFIL2

COMMAND:

3 (G) *** RPUNCH COMPLETE 00 00003 00394 00000 .RPUNCH00

COMMAND:

3 (H) *** RPUNCH COMPLETE 00 00003 00127 00000 .RPUNCH01

COMMAND:

3 (J) *** RPRINT COMPLETE 00 00001 00001 00000 LP01

COMMAND:  TERM
```

(B)140892(4 4)

APPENDIX E

EMULATOR MESSAGES

Table E-1 presents all of the messages displayed by the Emulator. The messages are organized in alphabetical order with an explanation following each message. Three asterisks precede certain messages when displayed; however, the asterisks have been removed for this table. If messages are displayed that are not in the table, they may be found in the *Model 990 Computer DX10 Operating System, Release 3, Reference Manual, Volume VI,* part number 946250-9706.

Table E-2 is a table of the completion/error codes that accompany several of the Emulator messages. The completion/error code indicates the status of the Emulator when the message is displayed. For completion/error codes not included in this list, consult the *Model 990 Computer DX10 Operating System, Release 3, Reference Manual, Volume VI,* part number 946250-9706.

Table E-1. Emulator Transaction Messages

Message	Description/Recovery Procedure
ACU NOT AVAILABLE	A CALL command and phone number have been entered on a system with an ACU configured, but the ACU is not available. Either a configuration error has occurred or the phone is off the hook. Correct the error and dial manually, or issue a DISC command and reenter the CALL command.
BELL RECEIVED	A BELL message has been received from the remote system.
BID ERROR	The remote system is not responding to the bid. Determine if the remote system is active or if it may have been disconnected. If disconnected, reconnect the link and retry the transmission.
CALL RETRY NECESSARY	A requested link connection (CALL or ANS) has terminated unsuccessfully without expiration of the retry count. If the call is being dialed manually, dial the number again; otherwise, no operator action is required.
COMM DEVICE NOT AVAILABLE	The Emulator has been generated with an incorrect address for the communication device, or the communication device is not working.

Table E-1. Emulator Messages (Continued)

Message	Description/Recovery Procedure

COMM TASK OPERATOR CONSOLE I/O ERROR

System initialization error. The operator console is not available and the Emulator is aborted. Correct the error and reinitialize the system.

COMMAND DEVICE NOT AVAILABLE

The device (or file) specified in response to the COMMAND ID: prompt is not available.

COMMAND/LOG FILE ASSIGN/OPEN ERROR

An error has occurred in attempting to assign or to open a command file or a log file. Ensure that the pathname is correct and reenter the command.

CONNECT FAILED

A requested link connection (CALL or ANS) has terminated unsuccessfully without expiration of the retry count. The command must be reentered to establish the desired connection.

CONNECT SUCCESSFUL

The communication link has been successfully made as requested via a CALL or ANS command.

IDLE ERROR

The user has specified the parameter Disconnect on Error and the Idle Threshold parameter has been exceeded. A disconnect is accomplished.

INSUFFICIENT MEMORY FOR COMMAND INPUT

The command buffer space available is less than that required to hold the command and its parameters. If the system is in the command file mode, the system issues this message and halts command input until buffer space becomes available, then it continues. If the system is in operator mode, the system issues this message and then issues a new COMMAND: prompt.

INVALID INPUT

The operator input just entered is incorrect. Check the spelling or see if the input is valid in context, and reenter.

Check whether the pathname is too long or contains invalid characters.

INVALID PARAMETERS

An invalid response has been made to one of the activation prompts.

Table E-1. Emulator Messages (Continued)

Message	Description/Recovery Procedure

I/O ERROR XX pathname

An error occurred during input from or output to the specified pathname. The XX is a system error code defined in Section IV of the *Model 990 Computer DX10 Operating System Reference Manual, Volume VI*, part number 946250-9706. The error codes described in this user's guide are Input/Output Errors and are displayed as 00XX.

LINE DISCONNECTED

The communication link has been successfully disconnected as requested or a disconnect has been sensed that was not requested.

LOG DEVICE NOT AVAILABLE

The log device specified in the response to LOG ID: is not available.

NO RECEIVE PATHNAME SPECIFIED

Data has been received for which there is no pathname specified in the receive queue.

OPEN ERROR XX pathname

The Emulator encountered an error in input from or output to the specified pathname. The XX is a system error code described in Section IV of the *Model 990 Computer DX10 Operating System Reference Manual, Volume VI*, part number 946250-9706.

OPERATOR MESSAGE QUEUE FULL

More than eight operator messages are queued for display on the operator console or log. The system halts message output until the queue has emptied. This condition should not occur unless the operator has stopped output and delayed restarting the output.

RECEIVED SEND REQUEST COMMAND pathname

A send request has been received and is to be processed.

SEND REQUEST COMMAND ERROR

A send request was received which could not be processed correctly due to an illegal pathname or no command buffers being available.

RPNAME COMMAND pathname

A receive pathname specification string has been received and entered in the receive queue for the receive device (RPR or RPU).

RPNAME COMMAND ERROR

A receive pathname specification string has been received but could not be processed correctly due to an illegal pathname or no command buffers being available.

Table E-1. Emulator Messages (Continued)

Message	Description/Recovery Procedure

RPRINT COMPLETE ZZ IIIII RRRRR 00000 pathname

or

RPUNCH COMPLETE ZZ IIIII RRRRR 00000 pathname

A data file reception (print or punch) has been completed to the specified pathname.

ZZ	=	communication completion/error code (see table E-2). If the completion/error code is not 00, an error has occurred. Refer to table E-2 for a list of the completion/error codes and their meanings.
IIIII	=	transaction ID of the queue entry just completed.
RRRRR	=	number of records transferred.
00000	=	number of records truncated.

TRANSACTION IN PROGRESS

An ABORT option specified a transaction ID for which a data transfer was in progress.

TRANSACTION NOT ON QUEUE

An ABORT option specified a transaction ID that was not present on any system queue.

TRANSMIT COMPLETE ZZ IIIII RRRRR 00000 pathname

A data file transmission has been completed from the specified pathname or text string (only the first pathname of multiple pathnames is printed).

ZZ	=	communication completion/error code (see table E-2). If the completion/error code is not 00, an error has occurred. Refer to table E-2 for a list of the completion/error codes and their meanings.
IIIII	=	transaction ID of the queue entry just completed.
RRRRR	=	number of records transferred.
00000	=	number of records truncated.

Table E-2. Communication Completion/Error Codes

Code	Description	Recovery Procedure
00	No error	None required.
18	Line disconnected	Connection must be reestablished if additional data transfers are desired.
51	I/O terminated abnormally	The receiving terminal received an EOT prior to receiving an ETX. The data file must be retransmitted to obtain a complete data transfer. Refer to Appendix F for a description of the EOT and ETX characters.
56	NAK threshold exceeded	The communication link is unusually bad. Try to reconnect the link to see if another circuit will allow transfers. Refer to Appendix F for a description of a NAK character.
57	ENQ threshold exceeded	The remote system is either responding incorrectly (wrong acknowledgement) or is not responding at all. Determine if the remote system has disconnected. Refer to Appendix F for a description of an ENQ character.

APPENDIX F

BINARY SYNCHRONOUS COMMUNICATIONS PROCEDURES

F.1 INTRODUCTION

The Emulator, like the IBM 3780 and 2780, uses the internationally recognized Binary Synchronous Communications (BSC) procedure to exchange data over telephone lines. The BSC procedure provides a standard framework for the communication exchange, its checking, and its correction. This framework is constructed from special data link control characters that are added and removed automatically by the Emulator so that files need only contain the data to be transmitted. The control characters supported by the IBM 3780 and 2780 and the Emulator are described in this appendix with a brief description of the circumstances in which they are employed.

Characters other than the data link control characters are also included in the data stream. These characters are the PAD character and the cyclic redundancy check (CRC) characters. The PAD character follows each transmission. Its function is to ensure that the last character of the transmission is sent prior to the transmitting station turning off. THe PAD character consists of eight "1" bits (hex FF).

Transmission errors are detected by the Emulator through a cyclic redundancy check performed on the data. Prior to transmitting, the transmitting station performs a division on the data and includes the result in the transmission. The numeric binary value of the data is divided by a constant. All characters that follow an STX character, except SYN but including ETB and ETX, are included in the binary value. In transparent mode, all characters between DLE STX and DLE ETX, including the ETX but not the DLE characters, are included in the value. For a DLE to be included in the value, it must be preceded by another DLE character. The remainder from the division is the cyclic redundancy check character and is inserted in the data stream following an ITB, ETB, or ETX character. The receiving terminal does a similar division upon reception of the data. If the receiving terminal's remainder matches the transmitting terminal's remainder, the transmission is accepted as error-free.

F.2 DATA LINK CONTROL CHARACTERS

The control characters supported by the Emulator are as follows:

SYN	Synchronous idle
STX	Start of text
ITB (IUS)	End of intermediate transmission block
ETB	End of transmission block
ETX	End of text
EOT	End of transmission
ENQ	Enquiry
ACK0 1	Alternating affirmative acknowledgement

NAK	Negative acknowledgement
DLE	Data link escape
WACK	Wait before positive acknowledgement
RVI	Reverse interrupt
TTD	Temporary text delay

F.2.1 SYN — SYNCHRONOUS IDLE. This character is used to establish and maintain synchronization of the transmission. Each message is preceded by at least two SYN characters that precede any control characters or data transmitted. Each message is followed by a PAD sequence.

F.2.2 STX — START OF TEXT. This character precedes a block of text and places the receiving station in the receive mode. STX is also the first character of a TTD sequence.

F.2.3 ITB (IUS) — END OF INTERMEDIATE TRANSMISSION BLOCK. The ITB (IUS) character is used to break up a long file transfer into convenient block sizes for error-checking purposes. It does not indicate an end to the transmission or a line turnaround. The ITB (IUS) character is always followed by the CRC character. The ITB (IUS) character is only transmitted by the 2780 Emulator. Transparent data containing ITBs (IUSs) must be transmitted in fixed-length records for the Emulator to receive it correctly.

F.2.4 ETB — END OF TRANSMISSION BLOCK. This character indicates the end of a block of data preceded by STX and indicates that another block follows. The CRC character for the block follows the ETB character. The ETB character forces a line turnaround and requires a response from the receiving station.

F.2.5 ETX — END OF TEXT. This character indicates the end of a block of data preceded by STX. It does not indicate that another block follows. The CRC character for the block follows the ETX character. The ETX character forces a line turnaround and requires a response from the receiving station.

F.2.6 EOT — END OF TRANSMISSION. This character is used to terminate a transmission and to indicate to the receiving station that it should return to the control mode. An EOT is also used to indicate certain error conditions. The DLE EOT sequence is used on a switched circuit to indicate to the receiver that the line should be disconnected.

F.2.7 ENQ — ENQUIRY. The ENQ character is used to bid for control of a line not currently active. If the response to the bid is ACK0, the receiving station has accepted the bid and transmission may begin. If a NAK response is returned, the requesting station may not transmit and must repeat its enquiry sequence.

The ENQ character may be used to request the retransmission of a reply when the anticipated reply is not received.

The ENQ character is also used in the TTD sequence.

F.2.8 DLE — DATA LINK ESCAPE. This character is used to provide supplementary line control characters such as WACK, ACK0/1, and RVI and to enable the transmission of control characters in transparent text. Transparent text sequences are initiated by the character combination DLE STX. Return to normal text mode occurs after either a DLE ITB, DLE ETB, or DLE ETX sequence.

F.2.9 ACK0/1 — ALTERNATING ACKNOWLEDGEMENT. ACK0 is an affirmative acknowledgement for line bids and even-numbered text blocks. It is composed of the character sequence DLE >70.

ACK1 is an affirmative acknowledgement for odd-numbered text blocks. It is composed of the character sequence DLE /.

F.2.10 NAK — NEGATIVE ACKNOWLEDGEMENT. This character is used to indicate that transmission errors have been detected in the last transmitted data block. The character is also used as a not-ready-to-receive response to a line bid (ENQ).

F.2.11 WACK — WAIT BEFORE TRANSMITTING POSITIVE ACKNOWLEDGEMENT. This is a positive acknowledgement sent by a receiving station when it is temporarily unable to receive more data blocks (see figure F-1). WACK is composed of the character sequence DLE ,. The 2780 Emulator cannot transmit the WACK character.

F.2.12 RVI — REVERSE INTERRUPT. This character is transmitted by a receiving station in place of an ACK to request that the transmitting station stop sending so that the receiving station may use the line to send priority data. The receiving station treats the RVI sequence as a positive acknowledgement, transmits any data remaining in its communication buffers, and releases the communication link (figure F-1). When the priority data has been transmitted, the original transmitting station again bids for the line and resumes transmitting where it was interrupted. RVI is composed of the character sequence DLE @. The Emulator cannot transmit the RVI character.

F.2.13 TTD — TEMPORARY TEXT DELAY. This character sequence is sent by the transmitting station when the station is not ready to transmit but does not want the line to be disconnected, as shown in figure F-2. The TTD sequence is:

SYN SYN STX ENQ PAD

The 2780 Emulator cannot transmit the TTD sequence.

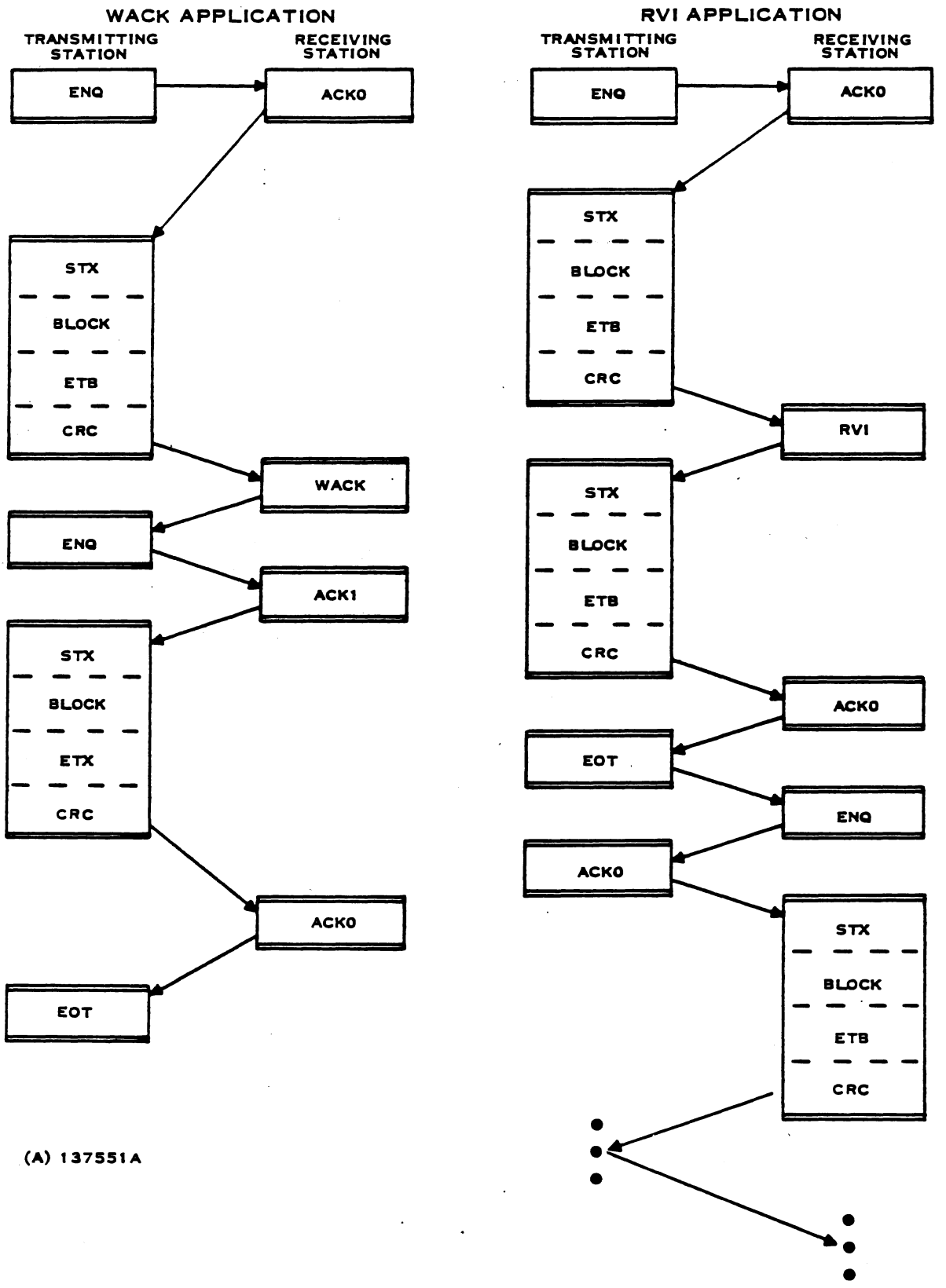


Figure F-1. WACK and RVI Applications

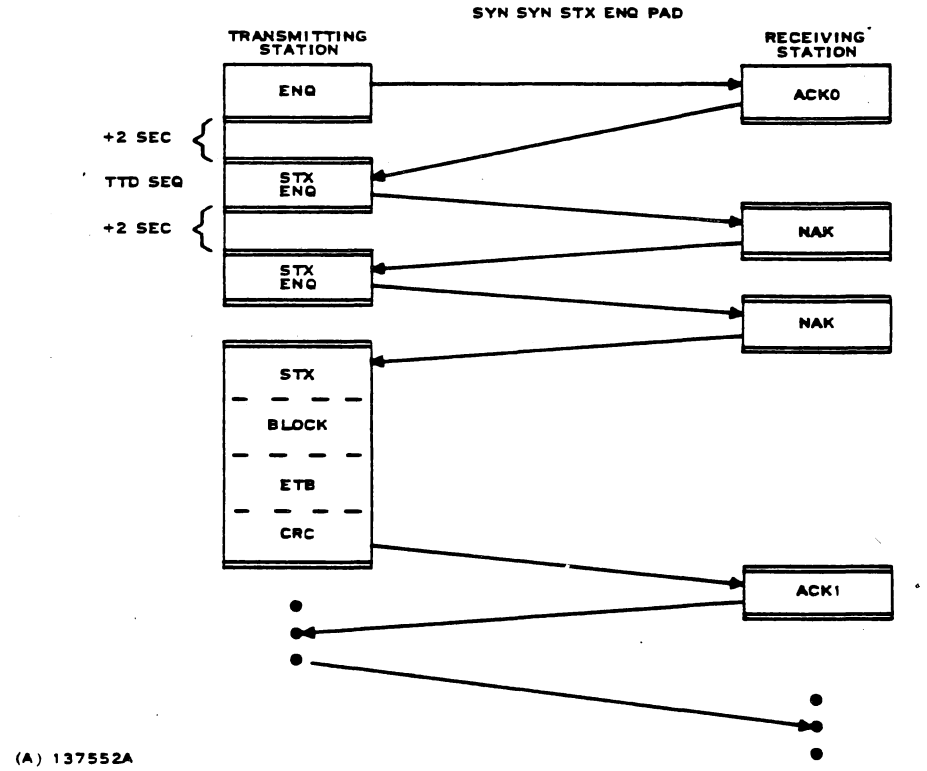


Figure F-2. TTD Application

APPENDIX G

ASCII — EBCDIC TRANSLATION TABLES

The IBM 3780 and 2780 protocol employed by the Emulator requires that the data be transferred as EBCDIC encoded data. The translation from the user data, prepared in ASCII code, to EBCDIC code is performed automatically by the Emulator so that line requirements are transparent to the user. Those EBCDIC characters less than >40 for which ASCII has no equivalent are translated to null characters; those EBCDIC characters >40 or greater for which ASCII has no equivalent are translated to space characters. Table G-1 gives ASCII to EBCDIC translation and table G-2 provides EBCDIC to ASCII translation. With transparency ON, the entire ASCII character set in table G-1 may be transmitted/received. With transparency OFF, the characters SYN, ENQ, ITB (US), IRS (RS), ETX, ETB, EM (for 2780), and GS (for 3780) may not be transmitted/received as data. Also, with transparency OFF, DC1, DC2, DC3, and ESC should not be transmitted as the first character of a data record because they will be interpreted by the receiving station as a device selection or as a vertical format control specification. To transmit/receive the entire EBCDIC character set in table G-2, the data must be transmitted/received in binary (990 to 990 communication only).

Table G-1. ASCII—EBCDIC Translation

ASCII Value	ASCII Character	EBCDIC Value	ASCII Value	ASCII Character	EBCDIC Value
00	NUL	00	18	CAN	18
01	SOH	01	19	EM	19
02	STX	02	1A	SUB	3F
03	ETX	03	1B	ESC	27
04	EOT	37	1C	FS	1C
05	ENQ	2D	1D	GS	1D
06	ACK	2E	1E	RS	1E
07	BEL	2F	1F	US	1F
08	BS	16	20	SPACE	40
09	HT	05	21	!	5A
0A	LF	25	22	"	7F
0B	VT	0B	23	#	7B
0C	FF	0C	24	$	5B
0D	CR	0D	25	%	6C
0E	SO	0E	26	&	50
0F	SI	0F	27	'	7D
10	DLE	10	28	(	4D
11	DC1	11	29	)	5D
12	DC2	12	2A	*	5C
13	DC3	13	2B	+	4E
14	DC4	3C	2C	,	6B
15	NAK	3D	2D	-	60
16	SYN	32	2E	.	4B
17	ETB	26	2F	/	61

Table G-1. ASCII—EBCDIC Translation (Continued)

ASCII Value	ASCII Character	EBCDIC Value	ASCII Value	ASCII Character	EBCDIC Value
30	0	F0	58	X	E7
31	1	F1	59	Y	E8
32	2	F2	5A	Z	E9
33	3	F3	5B	[	4A
34	4	F4	5C	\	E0
35	5	F5	5D	]	4F
36	6	F6	5E	^	5F
37	7	F7	5F	—	6D
38	8	F8	60	`	79
39	9	F9	61	a	81
3A	:	7A	62	b	82
3B	;	5E	63	c	83
3C	<	4C	64	d	84
3D	=	7E	65	e	85
3E	>	6E	66	f	86
3F	?	6F	67	g	87
40	@	7C	68	h	88
41	A	C1	69	i	89
42	B	C2	6A	j	91
43	C	C3	6B	k	92
44	D	C4	6C	l	93
45	E	C5	6D	m	94
46	F	C6	6E	n	95
47	G	C7	6F	o	96
48	H	C8	70	p	97
49	I	C9	71	q	98
4A	J	D1	72	r	99
4B	K	D2	73	s	A2
4C	L	D3	74	t	A3
4D	M	D4	75	u	A4
4E	N	D5	76	v	A5
4F	O	D6	77	w	A6
50	P	D7	78	x	A7
51	Q	D8	79	y	A8
52	R	D9	7A	z	A9
53	S	E2	7B	{	C0
54	T	E3	7C	¦	6A
55	U	E4	7D	}	D0
56	V	E5	7E	~	A1
57	W	E6	7F	DEL	07

Table G-2. EBCDIC—ASCII Translation

EBCDIC Value	EBCDIC Character	ASCII Value	EBCDIC Value	EBCDIC Character	ASCII Value
00	NUL	00	2A	SM	00
01	SOH	01	2B	CU2	00
02	STX	02	2C		00
03	ETX	03	2D	ENQ	05
04	PF	00	2E	ACK	06
05	HT	09	2F	BEL	07
06	LC	00	30		00
07	DEL	7F	31		00
08		00	32	SYN	16
09	RLF	00	33		00
0A	SMM	00	34	PN	00
0B	VT	0B	35	RS	00
0C	FF	0C	36	UC	00
0D	CR	0D	37	EOT	04
0E	SO	0E	38		00
0F	SI	0F	39		00
10	DLE	10	3A		00
11	DC1	11	3B	CU3	00
12	DC2	12	3C	DC4	14
13	TM	13	3D	NAK	15
14	RES	00	3E		00
15	NL	00	3F	SUB	1A
16	BS	08	40	SPACE	20
17	IL	00	41		20
18	CAN	18	42		20
19	EM	19	43		20
1A	CC	00	44		20
1B	CU1	00	45		20
1C	IFS	1C	46		20
1D	IGS	1D	47		20
1E	IRS	1E	48		20
1F	IUS	1F	49		20
20	DS	00	4A	¢	5B
21	SOS	00	4B	.	2E
22	FS	00	4C	<	3C
23		00	4D	(	28
24	BYP	00	4E	+	2B
25	LF	0A	4F	¦	5D
26	ETB	17	50	&	26
27	ESC	1B	51		20
28		00	52		20
29		00	53		20

Table G-2. EBCDIC—ASCII Translation (Continued)

EBCDIC Value	EBCDIC Character	ASCII Value	EBCDIC Value	EBCDIC Character	ASCII Value
54		20	80		20
55		20	81	a	61
56		20	82	b	62
57		20	83	c	63
58		20	84	d	64
59		20	85	e	65
5A	!	21	86	f	66
5B	$	24	87	g	67
5C	*	2A	88	h	68
5D	)	29	89	i	69
5E	;	3B	8A		20
5F	¬	5E	8B		20
60	-	2D	8C		20
61	/	2F	8D		20
62		20	8E		20
63		20	8F		20
64		20	90		20
65		20	91	j	6A
66		20	92	k	6B
67		20	93	l	6C
68		20	94	m	6D
69		20	95	n	6E
6A		20	96	o	6F
6B	,	2C	97	p	70
6C	%	25	98	q	71
6D	_	5F	99	r	72
6E	>	3E	9A		20
6F	?	3F	9B		20
70		20	9C		20
71		20	9D		20
72		20	9E		20
73		20	9F		20
74		20	A0		20
75		20	A1	~	7E
76		20	A2	s	73
77		20	A3	t	74
78		20	A4	u	75
79	`	60	A5	v	76
7A	:	3A	A6	w	77
7B	#	23	A7	x	78
7C	@	40	A8	y	79
7D	'	27	A9	z	7A
7E	=	3D	AA		20
7F	"	22	AB		20

Table G-2. EBCDIC—ASCII Translation (Continued)

EBCDIC Value	EBCDIC Character	ASCII Value	EBCDIC Value	EBCDIC Character	ASCII Value
AC		20	D6	O	4F
AD		20	D7	P	50
AE		20	D8	Q	51
AF		20	D9	R	52
B0		20	DA		20
B1		20	DB		20
B2		20	DC		20
B3		20	DD		20
B4		20	DE		20
B5		20	DF		20
B6		20	E0	\	5C
B7		20	E1		20
B8		20	E2	S	53
B9		20	E3	T	54
BA		20	E4	U	55
BB		20	E5	V	56
BC		20	E6	W	57
BD		20	E7	X	58
BE		20	E8	Y	59
BF		20	E9	Z	5A
C0	{	7B	EA		20
C1	A	41	EB		20
C2	B	42	EC	⊓	20
C3	C	43	ED		20
C4	D	44	EE		20
C5	E	45	EF		20
C6	F	46	F0	0	30
C7	G	47	F1	1	31
C8	H	48	F2	2	32
C9	I	49	F3	3	33
CA		20	F4	4	34
CB		20	F5	5	35
CC	⌐	20	F6	6	36
CD		20	F7	7	37
CE	⊔	20	F8	8	38
CF		20	F9	9	39
D0	}	7D	FA		20
D1	J	4A	FB		20
D2	K	4B	FC		20
D3	L	4C	FD		20
D4	M	4D	FE		20
D5	N	4E	FF		20

APPENDIX H

COMMAND SUMMARY

Command Name	Mnemonic	Parameters
Abort	ABORT	COM, TASK, transaction ID, ?
Answer	ANS	—
Block Length	BL	block length
Call	CALL	telephone number
Compression/Expansion	CE	ON, OFF, ?
Command File	CF	pathname
Conditional	IF	CON, DISC, S, RPR, RPU, ALLQ, ?
Disconnect	DISC	—
End of Media	EM	ON, OFF, ?
Log File	LF	pathname
Records Per Block	RB	number of records
Receive Pathname	RPN	ON, OFF, ?
Receive Printer Data	RPR	pathname
Receive Printer Default Pathname	RPRD	pathname
Receive Print Record Length	RPRRL	record length
Receive Punch DAta	RPU	pathname
Receive Punch Default Pathname	RPUD	pathname
Receive Punch Record Length	RPURL	record length
Reset	RESET	—
Send Data	S	pathname, text string
Send Data with Device Selection	SS	device and pathname, text string
Send Record Length	SRL	record length
Send Request	SRQ	ON, OFF, ?
Statistics	ST	CUR, CT, ALL, ALLQ, S, RPR, RPU, ?
Terminate Task	TERM	
Transparency	TY	ON, OFF, ?
Vertical Format Control	VFC	ON, OFF, ?
Wait	WAIT	CON, DISC, S, RPR, RPU, ALLQ, ?

ALPHABETICAL INDEX

INTRODUCTION

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along with the right side of the listing reference the following manual areas:

- Sections - References to Sections of the manual appear as "Section x" with the symbol x representing any numeric quantity.

- Appendixes - References to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.

- Paragraphs - References to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph is found.

- Tables - References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number:

  Tx-yy

- Figures - References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number:

  Fx-yy

PREFACE

The DX10 3780/2780 Emulators comprise a software package that runs on the 990/10 under the DX10 operating system and emulates the IBM 3780 and 2780 Data Communication Terminals. The IBM 3780 and 2780 Data Communication Terminals are nonprogrammable terminals that provide their users with remote input/output capabilities.

The DX10 3780/2780 Emulator software package consists of two emulators, the 3780 Emulator and the 2780 Emulator. Their features and operation are virtually the same, and for this reason, the features and operating instructions for both emulators are combined in this document.

This DX10 3780/2780 Emulator User's Guide describes the features and operation of the Emulators. The user's guide is divided into the following sections:

I General Description — Provides an introduction to the Emulators.

II Functional Description — Describes the features of the IBM 3780 and 2780 emulated by the DX10 3780/2780 Emulators.

III Functional Environment — Describes the functional environment of the Emulators.

IV Emulator Operation — Describes the activation of the Emulators and the commands used to operate the Emulators.

The following appendixes are provided for specialized and reference information:

A Emulator Hardware Requirements -- Describes the Emulators' hardware requirements.

B Emulator Software Requirements — Describes the Emulators' software requirements.

C Emulator Installation — Provides installation information.

D Communication Session — Provides an example of a communication session.

E Emulator Messages — Describes the Emulators' transaction messages and their meanings.

F Binary Synchronous Communications Procedures — Describes the Binary Synchronous Communications Procedures used by the Emulators.

G ASCII — EBCDIC Translation Tables — Provides ASCII to EBCDIC and EBCDIC to ASCII translation tables.

H Command Summary — Provides a summary of the Emulators' commands and their syntax.

This document is written for an operator with little or no data processing experience and for data processing personnel. Sections I, II, and III and Appendixes A through D, F, and G should be read by the person(s) responsible for establishing the operating environment. Section IV and Appendixes E and H should be read by the Emulator operator.

The following documents contain additional information related to the DX10 3780/2780 Emulators:

Title	Part Number
990 Computer Family Systems Handbook	945250-9701
Model 990 Computer DX10 Operating System Release 3 Reference Manuals, Volumes I-VI	946250-9701 through 946250-9706

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer DX10 3780/2780 Emulator User's Guide (946289-9701)

Manual Date: 15 February 1979 **Date of This Letter:** _____

User's Name: _____ **Telephone:** _____

Company: _____ **Office/Department:** _____

Street Address: _____

City/State/Zip Code: _____

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

CUT ALONG LINE

Location in Manual	Comment/Suggestion
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), TAPE AND MAIL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS GROUP
ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769

FOLD