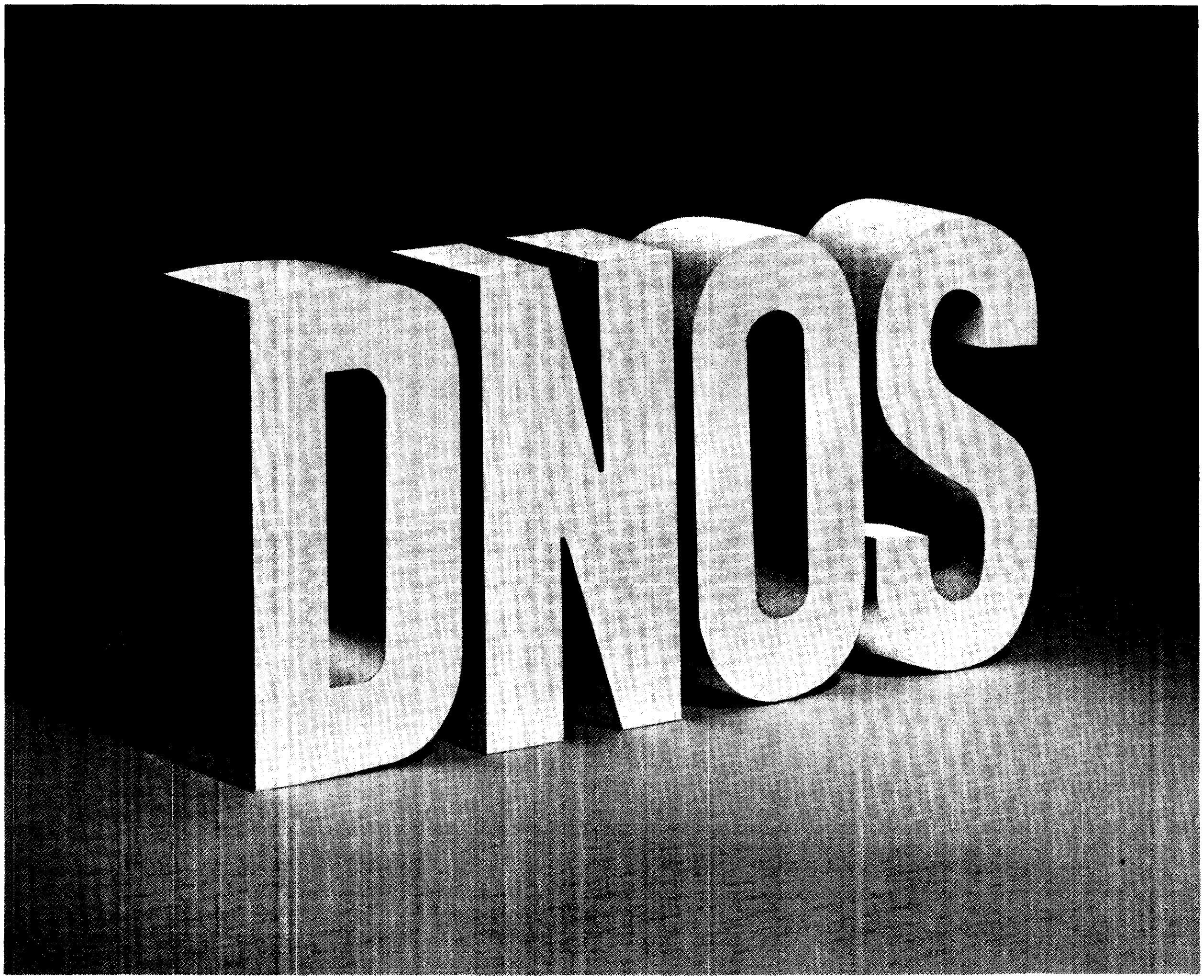
Model 990 Computer
DNOS 3780/2780 Emulator
User's Guide



Part No. 2270520-9701 *A
15 April 1982

TEXAS INSTRUMENTS
INCORPORATED

LIST OF EFFECTIVE PAGES

INSERT LATEST CHANGED PAGES AND DISCARD SUPERSEDED PAGES

Note: The changes in the text are indicated by a change number at the bottom of the page and a vertical bar in the outer margin of the changed page. A change number at the bottom of the page but no change bar indicates either a deletion or a page layout change.

Model 990 Computer DNOS 3780/2780 Emulator User's Guide (2270520-9701)

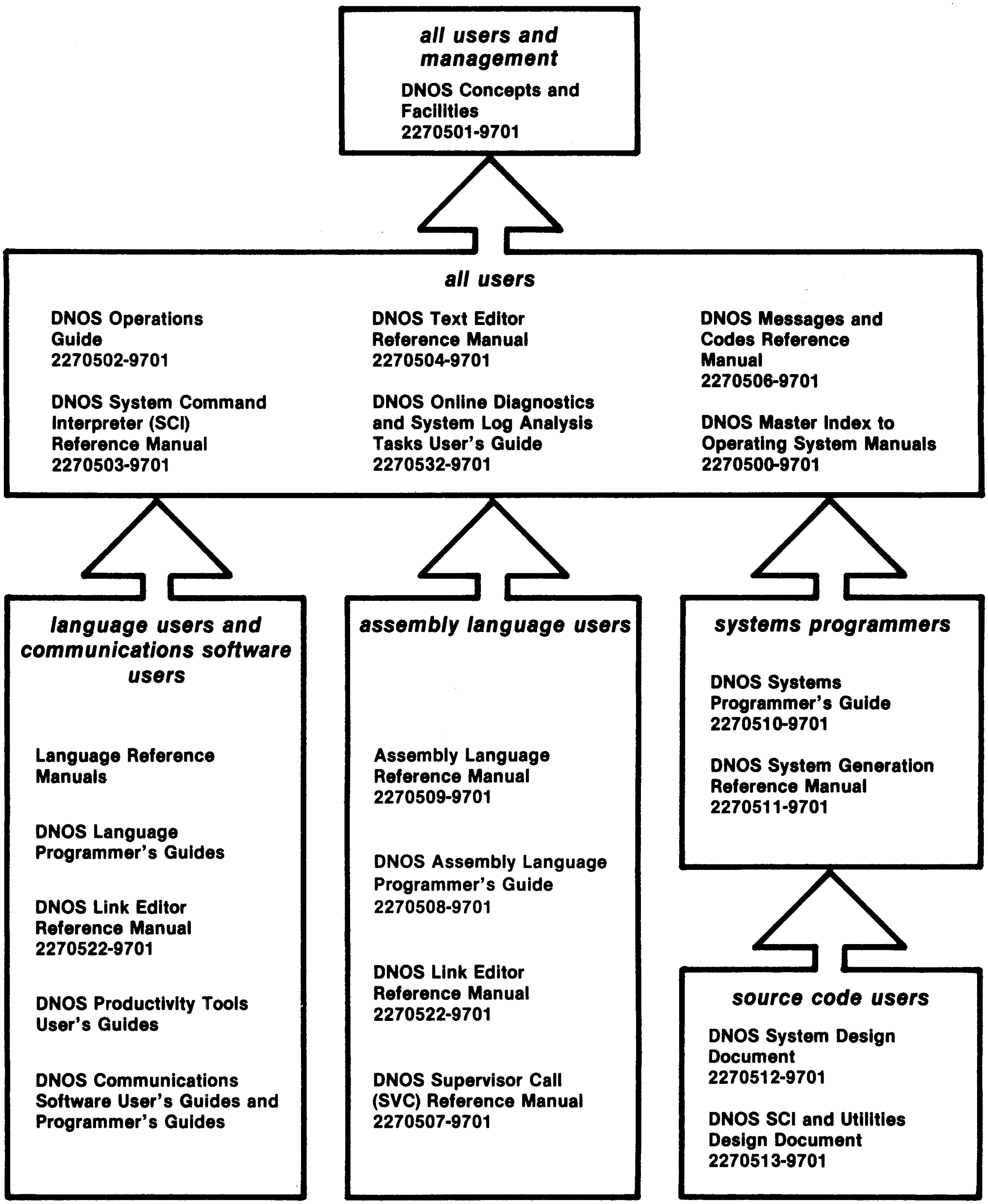
Original Issue .... 1 August 1981
Change 1 .... 15 April 1982

Total number of pages in this publication is 204 consisting of the following:

PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.	PAGE NO.	CHANGE NO.
Cover	1	4-38	0	C-1 - C-2	0
Effective Pages	1	4-39	1	C-3 - C-4	1
iii - xi	0	4-40	0	D-1	0
1-0 - 1-6	0	4-41	1	D-2 - D-10	1
2-1	0	4-42 - 4-48	0	E-1 - E-6	0
2-2	1	4-49 - 4-52	1	F-1	1
2-3 - 2-7	0	4-53	0	F-2 - F-16	0
2-8	1	4-54 - 4-55	1	G-1 - G-4	1
2-9 - 2-16	0	4-56	0	Index-1 - Index-3	0
3-1 - 3-9	0	4-57 - 4-59	1	Index-4 - Index-5	1
3-10	1	4-60	0	Index-6 - Index-12	0
3-11 - 3-14	0	4-61 - 4-63	1	Index-13	1
3-15 - 3-22	1	4-64 - 4-67	0	Index-14 - Index-16	0
4-1 - 4-4	0	4-68	1	User's Response	1
4-5 - 4-6	1	4-69 - 4-76	0	Business Reply	1
4-7 - 4-36	0	A-1 - A-6	1	Sales and Service	1
4-37	1	B-1 - B-6	1	Cover	1

DNOS
Distributed Network Operating System
Software Manuals

The manuals supporting DNOS are arranged in this diagram according to the type of user. The manuals most beneficial to your needs are those contained in the block identified as your user group and in all the blocks above that set.



all users and management

DNOS Concepts and Facilities
2270501-9701

all users

DNOS Operations Guide
2270502-9701

DNOS System Command Interpreter (SCI) Reference Manual
2270503-9701

DNOS Text Editor Reference Manual
2270504-9701

DNOS Online Diagnostics and System Log Analysis Tasks User's Guide
2270532-9701

DNOS Messages and Codes Reference Manual
2270506-9701

DNOS Master Index to Operating System Manuals
2270500-9701

language users and communications software users

Language Reference Manuals

DNOS Language Programmer's Guides

DNOS Link Editor Reference Manual
2270522-9701

DNOS Productivity Tools User's Guides

DNOS Communications Software User's Guides and Programmer's Guides

assembly language users

Assembly Language Reference Manual
2270509-9701

DNOS Assembly Language Programmer's Guide
2270508-9701

DNOS Link Editor Reference Manual
2270522-9701

DNOS Supervisor Call (SVC) Reference Manual
2270507-9701

systems programmers

DNOS Systems Programmer's Guide
2270510-9701

DNOS System Generation Reference Manual
2270511-9701

source code users

DNOS System Design Document
2270512-9701

DNOS SCI and Utilities Design Document
2270513-9701

2280078

DNOS
Distributed Network Operating System Software Manuals Summary

Concepts and Facilities
Presents an overview of DNOS with topics grouped into functions of the operating system. All new users (or evaluators) of DNOS should read this manual.

Operations Guide
Provides the information necessary to perform daily tasks at a TI 990 Computer installation using DNOS. Step-by-step procedures are presented for such tasks as operating peripherals, initializing the system, backing up the system, and manipulating disk files.

System Command Interpreter (SCI) Reference Manual
Describes how to use SCI in both interactive and batch jobs. Command procedures and primitives are described, followed by a detailed presentation of all SCI commands in alphabetical order for easy reference.

Text Editor Reference Manual
Shows how to use the Text Editor interactively on DNOS and includes a detailed description of each of the editing commands and function keys.

Messages and Codes Reference Manual
Lists the error messages, informative messages, and error codes reported by DNOS.

Online Diagnostics and System Log Analysis Tasks User's Guide
Provides the information necessary to execute the online diagnostic tasks and the system log analysis tasks and to interpret the results.

Master Index to Operating System Manuals
Contains a composite index to topics in the DNOS operating system manuals.

Programmer's Guides and Reference Manuals for Languages
Each programmer's guide describes one of the languages supported by DNOS (for example, assembly language, Pascal, COBOL). Each guide covers operating system information relevant to the use of that language in the DNOS environment. The details of the language itself, including language syntax and programming considerations, are in the language reference manual.

Link Editor Reference Manual
Describes how to use the Link Editor on DNOS to combine separately generated object modules to form a single linked output.

User's Guides for Productivity Tools
Each user's guide describes one of the productivity tools (for example, TIFORM, Query-990, DBMS-990, Sort/Merge) supported by DNOS. Each guide explains the function of the processor, its features, and its interface requirements.

User's Guides and Programmer's Guides for Communications Software
Describe the features, functions, and use of the communications software available for execution under DNOS. For example, there is a user's guide for the DNOS 3780/2780 Emulator software package.

Supervisor Call (SVC) Reference Manual
Presents detailed information about each DNOS supervisor call and general information about DNOS services.

Systems Programmer's Guide
Discusses the DNOS nucleus and subsystems at a conceptual and functional level and describes how to modify the system for a specific application environment.

System Generation Reference Manual
Contains the information needed to perform system generation, including pregeneration requirements, generation procedures, and information about postgeneration results.

System Design Document
Contains the information needed to understand the functioning of the system when using a source kit. This includes descriptions of the subsystems in detail, naming and coding conventions, module cross-references, data structure details, and information not found in other manuals.

SCI and Utilities Design Document
Presents design information about SCI and the DNOS utilities.

Preface

This document provides user information for the DNOS 3780/2780 Emulator software package. This package executes on a Model 990 Computer under the DNOS operating system and emulates the operation of the IBM 3780 Data Communication Terminal and the IBM 2780 Data Transmission Terminal. These IBM terminals provide their users with remote input/output (I/O) capabilities.

The DNOS 3780/2780 Emulator package consists of two Emulators, a 3780 Emulator and a 2780 Emulator. Since the features and operation of these Emulators are almost identical, this document combines appropriate information for both Emulators whenever possible. Discussions that refer to "the Emulator" apply to both Emulators.

This document is designed for data processing personnel responsible for establishing the operating environment for the Emulator. With guidance from the data processing personnel, the operator needs no significant experience with data processing or communications to successfully use the Emulator.

The User's Guide is divided into the following sections and appendixes:

Section

1 General Description — Provides an introduction to the Emulator.

2 Functional Environment — Describes the functional environment of the Emulator.

3 Functional Description — Describes the capabilities of the IBM 3780 and 2780 that are emulated by the DNOS Emulator and describes the features of the DNOS Emulator that enhance those capabilities.

4 Emulator Operation — Describes the activation of the Emulator and the commands that operate the Emulator.

Appendix

A Emulator Hardware Requirements — Describes the hardware requirements for the Emulator.

B Emulator Installation — Provides installation information.

C Communication Session — Provides an example communication session.

D Emulator Messages — Describes the Emulator's transaction messages and their meanings.

E Binary Synchronous Communications (BSC) Procedures — Describes the BSC procedures that the Emulator uses.

F ASCII/EBCDIC, EBCDIC/ASCII Translation Tables — Provides ASCII-to-EBCDIC and EBCDIC-to-ASCII translation tables for the United States, United Kingdom, Germany, France, Belgium, Sweden, Finland, Norway, Denmark, and Japan.

G Command Summary — Summarizes the Emulator commands and the syntax of each command.

The DNOS manuals shown on the frontispiece provide additional information related to this manual. Also, the following IBM documents contain information concerning the IBM 3780 Data Communication Terminal and the IBM 2780 Data Transmission Terminal:

Title	**Part Number**
Component Information for the IBM 3780 Data Communication Terminal	GA27-3063
Component Description: IBM 2780 Data Transmission Terminal	GA27-3005
General Information: Binary Synchronous Communications	GA27-3004

Contents

Paragraph Title Page

1 — General Description

2 — Functional Environment

3 — Functional Description

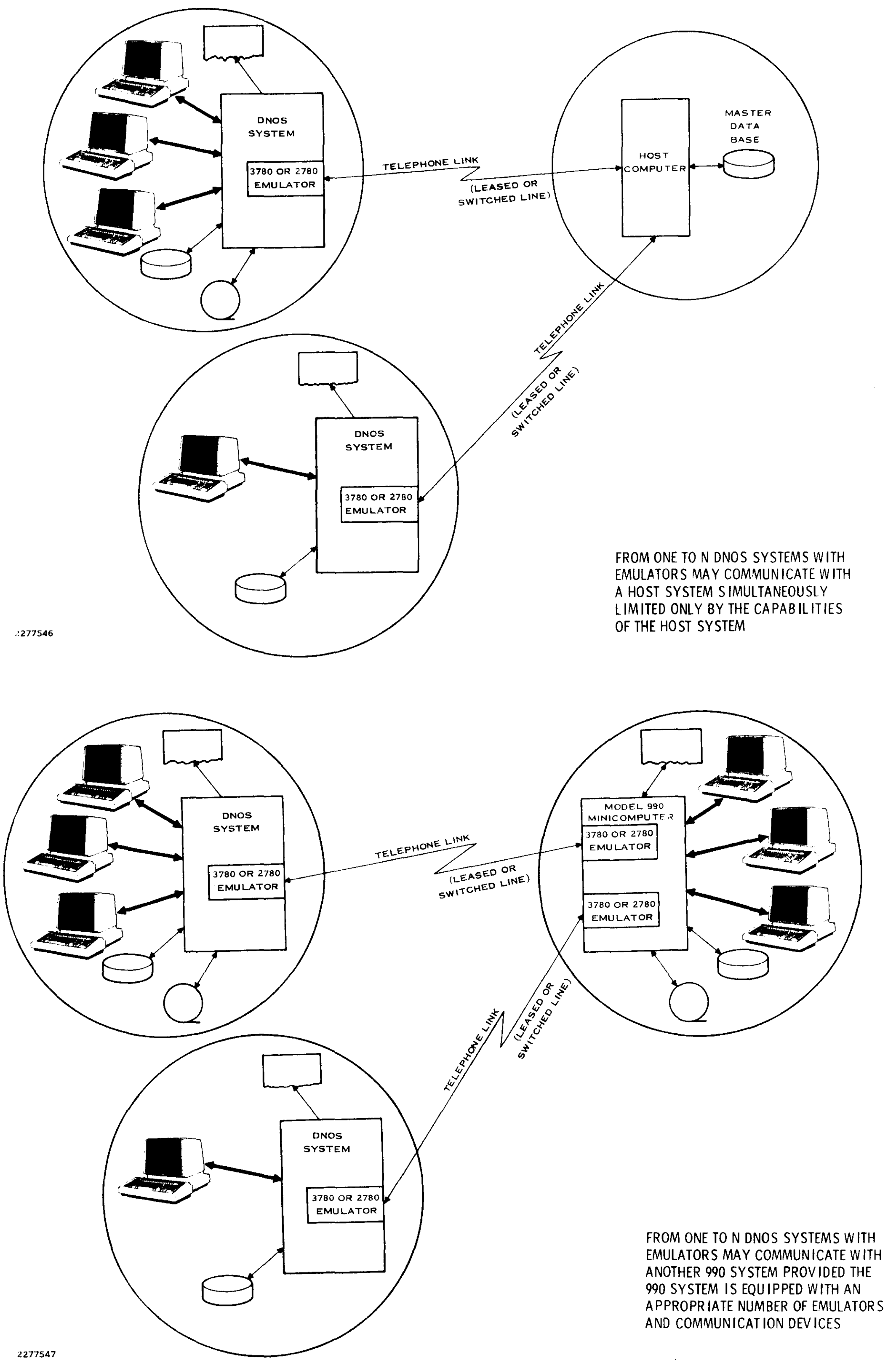
4 — Emulator Operation

Appendixes

Index

Illustrations

Tables



FROM ONE TO N DNOS SYSTEMS WITH EMULATORS MAY COMMUNICATE WITH A HOST SYSTEM SIMULTANEOUSLY LIMITED ONLY BY THE CAPABILITIES OF THE HOST SYSTEM

FROM ONE TO N DNOS SYSTEMS WITH EMULATORS MAY COMMUNICATE WITH ANOTHER 990 SYSTEM PROVIDED THE 990 SYSTEM IS EQUIPPED WITH AN APPROPRIATE NUMBER OF EMULATORS AND COMMUNICATION DEVICES

Figure 1-1. DNOS 3780/2780 Emulator Applications

1

General Description

1.1 INTRODUCTION

The DNOS 3780/2780 Emulators enable a Texas Instruments Model 990/10 or 990/12 Minicomputer using DNOS to communicate and exchange files with another computer over telephone lines (leased or switched, point-to-point). The DNOS 3780 and DNOS 2780 Emulators emulate the IBM 3780 Data Communication Terminal and the IBM 2780 Data Transmission Terminal, respectively. Data transmitted and received under the control of the DNOS 3780/2780 Emulator is identical to the data transmitted and received by an IBM 3780 or 2780. This enables DNOS, when emulating the IBM 3780 or 2780, to communicate and exchange files with any of the following:

- An IBM 3780 or 2780

- A host computer supporting the IBM 3780 or 2780 protocol

- Another emulator of the IBM 3780 or 2780 (including another 990 system equipped with a 3780 or 2780 Emulator)

Figure 1-1 illustrates some of the possible applications of the Emulator.

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

NOTE

> Since both DNOS Emulators have similar features and operating procedures, this manual addresses the DNOS 3780 and DNOS 2780 Emulators as a single emulator. Differences in their features and operation are called to the reader's attention through discussions designated as "3780 Emulator Users Only" or "2780 Emulator Users Only"; these discussions refer to "the 3780 Emulator" or "the 2780 Emulator" specifically. Otherwise, the 3780 Emulator and the 2780 Emulator are referred to as "the Emulator" and the discussion applies to both Emulators.

1.2 EMULATED IBM TERMINALS

The IBM 3780 Data Communication Terminal and the IBM 2780 Data Transmission Terminal are hardwired terminals that provide users with remote file-transfer capabilities when connected to an IBM host system that uses an appropriate access method (such as OS BTAM, OS TCAM, or DOS BTAM). Also, these terminals do not use stored program techniques, and the installed operator controls provide limited programming capability. The IBM 3780 and 2780 terminals consist of a controller, a console, two communications buffers, a card reader (used as an input device), and a line printer and/or a card punch (used as output devices). A modem (data set) compatible with the protocol and the transmission network is also required. Files transferred to and from the IBM systems are restricted to the input and output devices referenced above.

1.3 THE EMULATOR CONFIGURATION

The DNOS Emulator package executes on a Model 990 Minicomputer under DNOS. In addition to the minimum hardware requirements (see Appendix A), this system may be configured to include any or all of the following peripheral equipment:

- Disk media, including double-sided, double-density (DSDD) diskette
- Magnetic tape media
- Video display terminals (VDTs)
- Hard-copy data terminals, including the Texas Instruments Automatic Send/Receive (ASR) and Keyboard Send/Receive (KSR) terminals
- Line printers
- Card readers

Unlike the IBM 3780 and 2780, the Emulator can transfer files whose source and destination are not restricted to the card reader, card punch, and line printer. Any file or input device available to the system can supply data for a transfer, and any available file or output device may be specified to receive the information transferred.

Other tasks may execute on the DNOS system simultaneously with the Emulator task.

1.4 TERMINOLOGY

The Emulator package allows a system to communicate with another 990 computer equipped with a 3780/2780 Emulator, an IBM host system, an IBM 3780 or 2780 terminal, or any other system that emulates the IBM 3780 or 2780.

In this manual, the system in which the Emulator is installed is referred to as the *local station*. The system with which the Emulator is communicating is referred to as the *remote station* unless the Emulator is communicating with an IBM host system. In this case, the station with which the Emulator is communicating is referred to as a *host system*.

DNOS System Command Interpreter (SCI) commands activate the Emulator and provide auxiliary services. Emulator commands (added at installation time) initiate Emulator operations. To execute DNOS SCI commands, the CMD key is pressed to enter the command mode followed by the entry of the appropriate SCI command mnemonic. The Emulator command mnemonics may be entered after the Emulator has been activated and the COMMAND: prompt is displayed. In this manual any command not specifically designated as an SCI command is an Emulator command and is available after the Emulator is activated.

1.5 OPERATIONAL DESCRIPTION

The Emulator operates in an environment referred to as a *communication session*. The communication session begins when the operator activates (executes) the Emulator. During activation, the operator establishes the physical operating environment by specifying the following:

- Input device, file, or channel that issues commands
- Output device, file, or channel that receives the log of the transactions
- File that maintains the continuous journal of the transactions
- Communication device that is used to transmit and receive

After Emulator activation, a connection to a remote station may be established and file exchanges begin. Emulator commands initiate these operations and can be entered by the operator from an input device (operator mode) or can be issued from a previously prepared command file (command file mode). The commands specify the following:

- Data attributes for transmissions and receptions (that is, record lengths)
- Connection of a communication link
- Disconnection of a communication link
- Transmission of data
- Reception of data
- Use of special features

The Emulator can maintain a log of the transactions that occur during the communication session when this is specified by the operating environment. At various stages of operation, the log might be maintained on the operator console, any available output device, a relative record or sequential file, or an interprocess communication (IPC) channel, depending on the requirements of the communication session. In addition to the log, the Emulator can maintain a journal that is a continuous record of all Emulator activity; this is available for accounting and analysis purposes.

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

1.6 INSTALLATION

During installation, values are set to control several aspects of the Emulator operating environment. These values include the following:

- Time-out durations
- Retry counts
- Features enabled or disabled
- Block sizes
- Record sizes

The *Model 990 Computer DNOS 3780/2780 Emulator Object Installation Guide* describes these values in detail. Appendix B contains tables that summarize these values and provide an area where the user-defined values may be recorded. The operator can change many of these values during the operation of the Emulator by using Emulator commands.

1.7 FEATURES AND CAPABILITIES

The following paragraphs briefly describe 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. Section 3 contains a detailed description of each of these features.

1.7.1 Emulated IBM Features and Capabilities

The emulated features and capabilities are as follows:

- Data link control — Binary Synchronous Communications (BSC) procedures control point-to-point data exchanges.
- Terminal identification — The Emulator can optionally identify itself to the host or remote system with an identification character sequence; this sequence may be exchanged with the remote station with each connection. The Emulator may optionally verify the terminal identified. The maximum length allowed for the terminal identification sequence is 20 characters.
- Switched network control — The Emulator can automatically answer or disconnect when the communication line is attached to a switched (dial-up) network and is equipped with an appropriate auto-answer modem. The Emulator also can direct a remote station to disconnect.
- 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 (ENQ) retry — The Emulator can retry a bid for the communication line an infinite number of times.

- Input/Output (I/O) devices — The Emulator can use any available system resource for input and any available system resource for output. Although the card punch is not supported, the Emulator transmits and receives data formatted for the card punch. (The operator can direct received card punch data to any available file or device.) Optionally, the operator may use the DNOS spooling technique to spool to a device. As a result, the device can be logically assigned to an output device or class of devices.

- Horizontal format control (HFC) and vertical format control (VFC) — The Emulator formats printer output by translating the format control characters to printer format codes.

- Two principal modes of operation — Two modes of operation are line mode and home mode. The Emulator performs the line mode operations of transmitting or receiving data over the communication lines (for example, local card reader to remote printer operations). In contrast, DNOS utilities (not the Emulator) perform the home mode operations (for example, local card reader to local printer operation) utilizing the Copy/Concatenate (CC) or Print File (PF) SCI commands.

- Space compression/expansion (3780 Emulator users only) — The Emulator can replace strings of spaces with an information-gap-separator (IGS) character and a space-count character to reduce the number of spaces transmitted, thereby improving transmission efficiency.

- End-of-media (EM) (2780 Emulator users only) — The Emulator can place an end-of-record indicator at the end of transmitted records to truncate trailing blanks, thereby improving transmission efficiency.

- EBCDIC transparency — The Emulator can transmit control characters as data characters.

- Binary data transmission/reception — The Emulator can transmit and receive an entire 256 eight-bit (>00 to >FF) character set by specifying that the data be handled as binary data (that is, no translation or back translation).

- Component selection — The local station can select the output device at the remote station.

- Audible alarm — Logged messages can be displayed or printed on the operator console to take the place of an audible alarm and alert the operator that the station requires attention.

- Multiple record transmission — The 3780 Emulator can transmit as many records per block as can fit in the specified block. The 2780 Emulator can transmit either a fixed number of records per block or as many records per block as can fit in the specified block.

1.7.2 Additional Emulator Features and Capabilities

Additional features and capabilities of the Emulator are as follows:

- International character sets — The Emulator supports international character sets that include unique characters for the United Kingdom, France, Belgium, Germany, Sweden, Finland, Norway, Denmark, and Japan.

- Command file operation — The Emulator can be operated by a sequential file or a relative record file containing predefined Emulator commands or by an IPC channel that processes commands from tasks executing on DNOS. These modes of operation do not require the presence of an operator at the local station.

- Receive pathname feature — The local station can specify the destination pathname for transmitted data.

- Send request feature — The local station can send a message to the remote station requesting that the remote station transmit a specified file or portion of a file to the local station.

- Bid task feature — Two bid task features can be requested:

  - The local station can send a message to the remote station requesting that the remote station execute a specified task.

  - The local station can execute a DNOS task on the local DNOS system.

- Partial file data transmission — The Emulator can begin transmission at specified records within a file to allow efficient recovery from error situations that result in incomplete transmissions.

- Journal feature — The Emulator can maintain a sequential record of all Emulator activity for accounting or analysis.

- Operator message feature — The local station can send a message to the remote station's log and/or journal.

- Automatic call unit (ACU) feature — The Emulator command source can initiate a communication line connection without a manual dial-up operation.

1.7.3 IBM Features and Capabilities Not Emulated

The features and capabilities not emulated are as follows:

- 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 3780 and 2780 can use data sets having no clocking signals, since a synchronous clock feature provides them with an internal clock.

- Multipoint data link control — The IBM 3780 and 2780 employ multipoint data link control to operate on a communication line with multiple stations.

2

Functional Environment

2.1 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 operator activates (executes) the appropriate Emulator (see Section 4). During Emulator activation, the operator must specify the following:

- Primary command source
- Primary log destination (optional)
- Journal destination (optional)
- Communication device name

The primary command source must be specified in response to the COMMAND ACCESS NAME: prompt displayed when the Emulator is activated and may be any one of the following:

- Video display terminal (VDT) or a hard-copy display terminal used by the operator (operator mode)
- Input device, sequential file, or relative record file (command file mode)
- Interprocess communication (IPC) channel (command file mode)

The primary log destination is optional. The operator can specify the destination in response to the LOG ACCESS NAME: prompt. When selected, it can be maintained on one of the following:

- VDT
- Hard-copy display terminal
- Output device
- Sequential or relative record file
- IPC channel

The journal is also optional and is specified by the operator in response to the JOURNAL ACCESS NAME: prompt, which is displayed when the Emulator is activated. When selected, it can be maintained on one of the following (although output to a sequential or relative record file is usually desired):

- VDT
- Hard-copy display terminal
- Output device
- Sequential or relative record file
- IPC channel

The desired communications channel (controller, modem, and line) must be designated in response to the COMM DEVICE NAME: prompt. The name of each communications channel attached to the system is assigned at the time the communications system is installed.

During the communication session, the Emulator issues command prompts to which the operator or the designated command source must respond. These responses initiate the Emulator transactions. The Emulator performs each transaction and then issues messages that indicate the status of the transaction (for example, completed successfully, completed unsuccessfully, or error encountered). While these transactions are taking place, the Emulator maintains a log of all transactions and messages. The log is displayed on or written to a destination specified during Emulator activation; this can be changed by the operator (or command file) during Emulator operation, as appropriate. Also, a journal containing a continuous record of all activity can be maintained; however, *the journal's destination cannot be altered during Emulator operation*. The communication session ends when the Emulator task is terminated.

2.2 OPERATIONAL MODES

Emulator commands initiate Emulator transactions. An operator, a sequential or relative record file, an input device, or an IPC channel can issue the Emulator commands. During Emulator activation, the operator must specify the command source in response to the COMMAND ACCESS NAME: prompt. This specifies the command source for the session. The command source selected at activation is called the *primary command source*. Any command source initiated after activation is called the *secondary command source*. The primary command source determines the initial mode of operation for the session as follows:

- Operator Mode — The operator enters commands on a VDT or hard-copy display terminal to direct Emulator activity interactively.
- Command File Mode — The Emulator receives its commands from one of the following:
  - A sequential or relative record file
  - An input device
  - An IPC channel

An IPC channel is a DNOS intertask communications facility. In this instance, the IPC channel allows a DNOS task and the Emulator task to communicate directly. The DNOS task writes commands to a channel; independently, the Emulator task reads those commands from that channel. Similarly, the DNOS task reads messages that the Emulator task writes to a channel.

During the operation of the Emulator, the primary command source can issue a Command File (CF) command to specify that a secondary command source will issue further commands. The pathname entered in response to the CF command prompt identifies the secondary command source. For each activation the Emulator has only one primary command source (that is, the one specified when the Emulator is activated). This primary command source remains open until Emulator termination. However, the Emulator can have several secondary command sources.

The CF command opens the specified input file, device, or IPC channel which then becomes the active command source. If this secondary command source subsequently issues a CF command, the active secondary command file or channel closes and the newly named command file or channel opens. The session continues in command file mode until all commands in the last secondary command source have been executed and control returns to the *primary command source*, or until the Emulator terminates.

2.2.1 Operator Mode

The Emulator enters the operator mode when the response to the COMMAND ACCESS NAME: prompt (displayed when the Emulator is activated) is the name and number of an interactive terminal or the default ME (which specifies the interactive terminal at which activation is taking place). When the Emulator is in the operator mode, the operator uses a VDT or hard-copy display terminal to issue commands, respond to command prompts, and react to console messages. The prompts, responses, and Emulator messages are logged on the terminal display. The Log File (LF) command can also be used to direct the log to a relative record or sequential file or to an IPC channel. *When the log is directed to a file, an output device, or an IPC channel, all prompts and Emulator messages continue to be displayed at the operator's terminal.* Refer to paragraph 4.4.1 for further information about Emulator operation in the operator mode.

2.2.2 Command File Mode

A command file is a prepared set of responses to the command prompts. These responses can be stored as records in a sequential or relative record file or can be issued from a program attached to an IPC channel. The sequential file or relative record file must be created by using the DNOS Text Editor prior to Emulator activation. The IPC channel must be created by using the SCI command Create the IPC Channel (CIC) prior to Emulator activation.

The Emulator enters command file mode when the response to the COMMAND ACCESS NAME: prompt (displayed when the Emulator is activated) is the access name of a command file. This command file is the primary command source. The Emulator also enters command file mode when the operator issues a CF command while the Emulator is in the operator mode. The command file specified by the CF command is the secondary command source.

An IPC channel is used when an application requires that an application task interact with the Emulator. The records generated by the application task (identical to the records read from a sequential or relative record file) contain the commands and command parameters to be used in response to the Emulator command prompts.

The application task issues commands to the Emulator by writing them to a predefined IPC channel. The Emulator reads the commands, takes appropriate action, and responds to the application task by writing messages (when necessary) to another IPC channel. The application task reads and interprets each message and, from this, determines the subsequent sequence of commands directed to the Emulator via the IPC channel specified as the command file.

Using operations similar to I/O operations, DNOS provides communication between tasks through IPC channels. Each of the tasks bound in an IPC channel link is analogous to a peripheral device; the IPC channel is associated with each task and has a chronologically ordered pending request queue. The DNOS task writes commands to and reads them from individual symmetric channels. *Each IPC channel is created by using DNOS SCI commands or supervisor calls (SVCs)*; is defined as a symmetric, exclusive, job-local channel; and is accessed by a channel pathname.

While in command file mode, the command file commands direct the session until the Emulator terminates or a secondary command source is specified. The operator can direct the log of the transactions to the terminal in use, to a relative record or sequential file, or to an IPC channel. Refer to paragraph 4.4.2 for a discussion of Emulator operation during command file mode.

2.3 RECORDS OF THE SESSION

Records of the communication session, a log and a journal, are made while the Emulator is operating. The operator may optionally specify the log and/or journal during Emulator activation. *The log includes Emulator prompts, operator responses, command file commands, and Emulator message output*. The journal is a continuous record of the communication session. In some instances, the records of the session can be split between the local and remote stations.

When the send request feature is used, the record of the session is shared between the two Emulators according to their participation in the data exchange. For example, the Emulator transmitting the request logs the prompts, responses, and messages associated with the transmission; the Emulator receiving the request and making the requested transmission logs the prompts, responses, and messages associated with the reception and the transmission. Each Emulator outputs the logged information to the file, device, or channel that is currently being used for log output.

Each time a log file or journal file is initialized, the appropriate string is written to it, as follows:

```
(date/time)-LOG FILE 2780 ASSIGNED-RID=(run ID)-COMM=(device name)
```

or

```
(date/time)-LOG FILE 3780 ASSIGNED-RID=(run ID)-COMM=(device name)
```

This identification of each separate log file is helpful to users having applications that require the interpretation of log file data.

2.3.1 Emulator Log

The operator's response to the LOG ACCESS NAME: prompt during Emulator activation determines the primary log destination. The primary log destination may be one of the following:

- Device. The log can be directed to any available output device, including the operator's VDT or hard-copy display terminal. A log directed to a VDT cannot be printed.

- Relative record or sequential file. To log to a relative record file, the operator must create the file prior to Emulator activation. The Emulator automatically creates a sequential file when a nonexistent pathname is entered in response to the LOG ACCESS NAME: prompt (displayed when the Emulator is activated) or in response to the PATHNAME: prompt of the LF command prompt set. When the log file is a relative record file or a sequential file, the operator can use the Show File (SF) SCI command to view the file while the Emulator is writing to it.

- IPC channel. A log directed to an IPC channel can be printed only if a task is programmed to read the channel and print the information read. The operator must have created the IPC channel prior to Emulator activation.

Using the LF command, the operator can change the log destination or designate a new log through the primary command source or secondary command source. When the LF command is issued, the writing of log information to the primary log destination ceases, and writing to the secondary log destination begins. However, when the primary command source is a terminal, the operator can specify a second log destination; the log is output to both that destination and the terminal. (See Section 4 for a discussion of the consequences of specifying the command terminal as the log device.) If for any reason data cannot be written to the log file, the current message is written to the journal file, if one is specified. If no journal exists, an error message is logged to the system log file.

2.3.2 Emulator Journal

The operator can specify a journal during Emulator activation in response to the JOURNAL ACCESS NAME: prompt. The journal serves a purpose similar to that of the computer system log file in that it provides a record of all Emulator activity for use in accounting or analysis.

If a journal is specified, all information destined for the log file is also logged to the journal file, regardless of multiple LF commands from the command source. The sequential journal file is automatically created when the Emulator is activated. If the journal file cannot be opened, an error message is written to the screen of the activating terminal to alert the operator. If the operator does not enter a pathname in response to the JOURNAL ACCESS PATHNAME: prompt, no Emulator journal is maintained.

If for any reason data cannot be written to the journal file, the current message destined for journal logging is aborted and an error message is logged to the system log file. Assigning the journal and log to the same pathname during activation causes the following error message to be logged to the operator's terminal:

```
(date/time) 1026 INVALID JOURNAL/LOG ACCESS NAMES
```

If the log file is directed to the journal file pathname by the LF command after activation, the request is aborted and the following error message is logged to the current log file:

```
(date/time) 1008 COMMAND/LOG FILE ASSIGN/OPEN ERROR
```

2.4 EMULATOR COMMAND PROMPTS

Command prompts are messages that guide the operator in supplying the information required for the operation of the Emulator. The prompts are either displayed on the operator's terminal or logged to a file, channel, or device.

In operator mode, the command prompts and messages appear on the operator's terminal. When the Emulator issues a COMMAND: prompt, the operator may respond in general or condensed format. When using the general format, the operator enters a command in response to the prompt. If the command has parameters or options associated with it, the Emulator issues prompts requesting each parameter or option. To display all possible legal replies to a specified prompt, the operator must enter ? in response to the prompt. When using the condensed format, the operator must respond to the COMMAND: prompt with a command; if the command requires parameters or options, the required parameter(s) or option(s) is appended to the command with asterisks. Unless one of the responses is incorrect, no more prompts are issued for that command.

In command file mode or IPC mode, the command source is prepared with predetermined responses to the command prompts. These responses can be in either general or condensed format and require no interactive prompting.

Section 4 describes the parameters or options available with each command. Table 2-1 lists the Emulator requirements for each command prompt.

Table 2-1. Command Prompts

Prompt	Command Source Requirements
COMMAND:	The Emulator requires an Emulator command from the command source.
DEVICE:	The Emulator requires the command source to respond with one of the following parameters to designate an output device at the remote station: PR (for printer) or PU (for punch)
PATHNAME:	The Emulator requires the command source to respond with a legal pathname or synonym (see paragraph 4.3). The send data commands, S and SS (see paragraph 4.5), also allow a text string enclosed in < > (that is, <text string>).
LENGTH:	The Emulator requires the command source to specify a record or block length (in bytes).
NUMBER:	The Emulator requires the command source to respond with a hexadecimal or decimal number as a parameter. The command source must precede hexadecimal numbers with the character > or a leading zero.

Table 2-1. Command Prompts (Continued)

Prompt	Command Source Requirements
OPTION:	The Emulator requires the command source to specify an option (described in Section 4) or indicate whether a feature is to be enabled or disabled (ON or OFF).
SEQUENCE:	The Emulator requires the command source to respond with a terminal identification character sequence (see paragraph 4.5).
PROGRAM:	The Emulator requires the command source to respond with the name of a valid program name contained in the DNOS .S$SHARED program file. The program name can be from one to eight characters in length. The first character must be alphabetic, and the remainder can be alphabetic or numeric.
STATION:	The Emulator requires the command source to respond with the identification (device class and number) of the interactive terminal at which the task (already specified) is to be executed.
EXECUTE:	The Emulator requires the command source to respond with a YES or NO. A YES response causes a task (already identified) to execute concurrently with the Emulator; a NO response suspends the Emulator until the task terminates.
PARAMETERS:	The Emulator requires the command source to respond with up to four ASCII characters to specify parameters to be passed to a bid task.

2.5 EMULATOR COMMANDS

During the communication session (from Emulator activation through Emulator termination), the Emulator operations consist of issuing commands and responding to prompts or messages. The commands can be categorized as follows:

- Change default commands
- Connect commands
- Send data commands
- Receive commands
- Command file commands
- Disconnect commands
- Miscellaneous commands

The operator can use change default commands to change installation parameters. Default values are set during installation to specify the following:

- Block length
- Receive print record length
- Receive punch record length
- Send record length
- Records per block (for 2780)
- Default pathname for the printer
- Default pathname for the punch
- Features enabled (ON) or disabled (OFF)
- Terminal ID sequence

The change default commands enable the operator to change many of the values set during installation for the duration of the current communication session. The communication session includes all operations during execution of the Emulator. These commands are often issued prior to transmitting or receiving since they set the specifications for those transmissions and receptions occurring immediately following the execution of the commands. Refer to Appendix B for a list of default and standard values that are set during Emulator installation and that can be changed by appropriate commands after Emulator activation.

The connect commands place a telephone call or answer a call to establish a telephone line connection.

The send data commands are used to send data. One send command can send as many files as the queue buffer space allows, as defined during installation.

The receive commands specify the pathname of the file in which the data received from the remote station is to be received.

The command file commands are used to enter command file mode and to control the execution of a command file. The operator can use a command file anytime during a communication session as a secondary command source; an alternative is to specify the command file as the primary command source for the session. The log file command is used optionally to direct the log output to an output device, IPC channel, or file during Emulator execution.

The disconnect commands break the telephone connection. Two of these commands, the Abort and Terminate Task commands, can also terminate the Emulator.

The miscellaneous commands reset and idle the communications line, display the statistics for the session, and assign a local station identification character sequence.

2.6 DATA TRANSMISSION

A data transmission is initiated by a send command issued by any of the following: the operator, a command file, an IPC channel, or a send request. The send commands are entered on the send queue, which is an ordered queue of all send commands in the order of their entry (first-in, first-out). One send command can send multiple files or text strings; the number of files or strings sent is limited only by the queue buffer limit assigned in response to the PENDING REC/SEND REQUESTS: prompt during installation. A send command entered on the send queue, is called a *transaction*, and is assigned a unique identifying number called a *transaction ID*.

Using the pathname specified in the send command, the Emulator "opens" the pathname and begins transferring data to an input buffer. From the device/file input buffer, the data is translated (or not translated, as 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) in the last file of the send command is reached, an irrecoverable error occurs, or a processor interrupt is received from the host system. Figure 2-1 illustrates the buffering of data transmissions.

Occasionally, transmissions may be interrupted. Either the transmitting station or the receiving station can initiate the interruption through commands, physical disconnections, communication link errors, or system errors. One command that interrupts a transmission is the Abort command. When ABORT*TASK or ABORT*COM is issued, transmissions in progress are aborted and deleted from the queues. An Abort command can also clear one or all of the queues. ABORT*S or ABORT*ALLQ clears the send queue; ABORT*ALLQ clears the send and receive queues. Another command that interrupts a transmission is the RESET command. It causes the receiving station to transmit an end-of-transmission (EOT) signal at the end of the next received block. The EOT resets the communications line to the idle line condition.

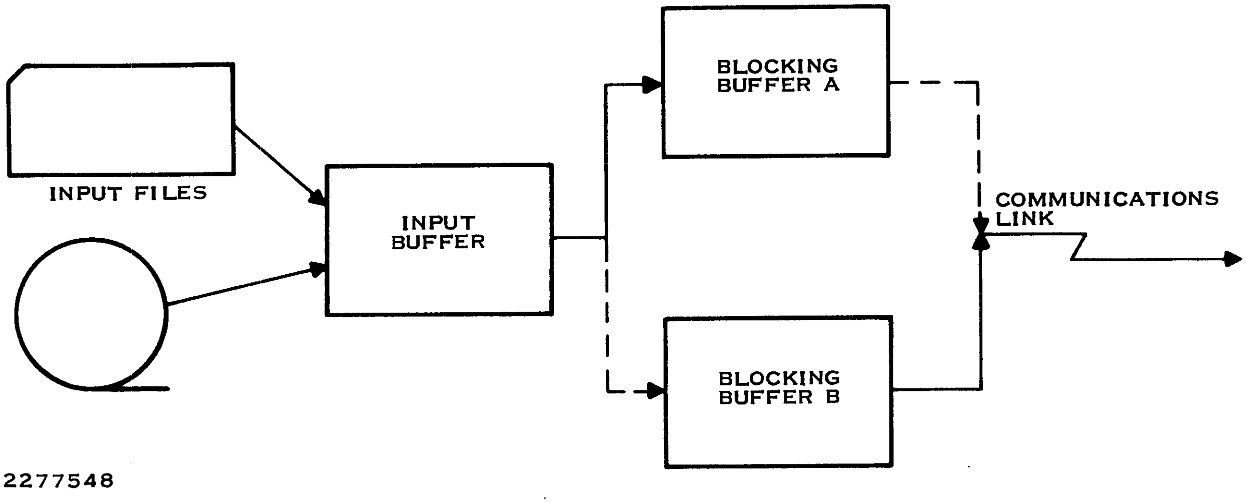


Figure 2-1. Data Transmission Buffering

To interrupt a local station's transmissions, the host system can respond by issuing a Reverse Interrupt (RVI). This instructs the transmitting station to temporarily cease transmitting so that the host system can transmit data of a higher priority. When the host completes its transmission, the transmitting station automatically resumes transmission at the point of interruption.

A physical disconnection initiated erroneously by either station or occurring in the physical path between the stations is treated as a communication link error. The result of a communication link error depends on whether it occurs during or prior to a transmission:

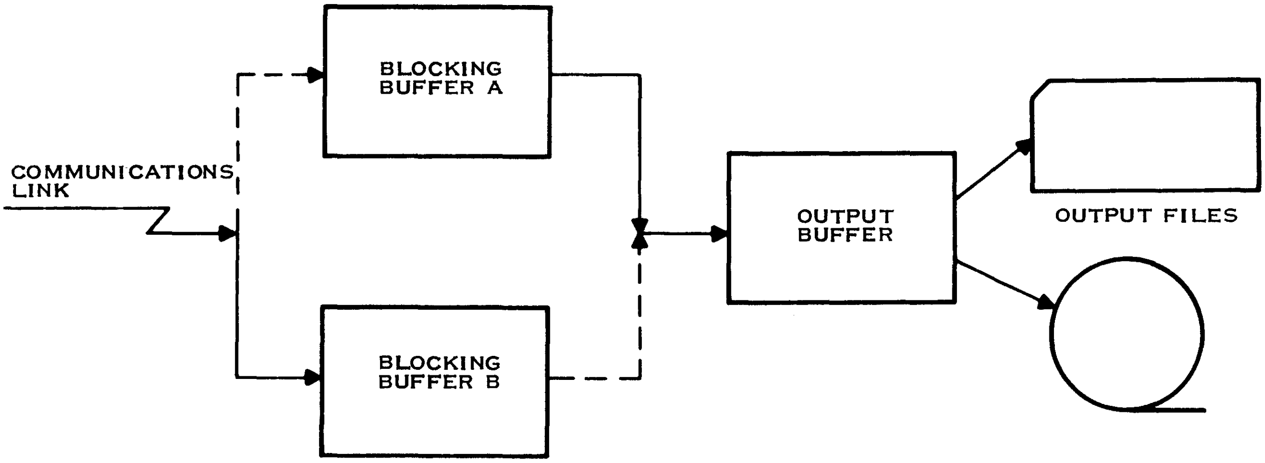
- If during a transmission, the transaction is aborted and deleted from the queue, and the next transaction in the queue is initiated.

- If prior to a transmission, the transmission is left in the queue and is transmitted next when the connection is reestablished.

When a system error (usually an I/O error or an open error) occurs during a transmission or prior to a transmission, the transmission is aborted and deleted from the queue, and the next transmission in the queue is processed.

2.7 DATA RECEPTION

The remote station bidding for the communication link initiates data reception. When the Emulator is not transmitting during communication sessions, it assumes a receive state and waits for incoming data. When a bid is received, the Emulator accepts it, receives data from the line, and writes the data to the alternating communication buffers. From there it is transferred to an intermediate buffer for system output (Figure 2-2). Data recognition is validated using the transmitted Cyclic Redundancy Check (CRC) characters (refer to Appendix E), and retransmissions are automatically requested if the CRC fails. The receiving station specifies the destination of received data by using the Receive Printer Data (RPR) and Receive Punch Data (RPU) commands. When one of these commands is executed, the pathname specified is entered in one of the two receive queues (receive print or receive punch). Once entered, the RPR or RPU is a transaction and is assigned a unique transaction ID.



2277549

Figure 2-2. Data Reception Buffering

The receive queues use the following order for queuing:

```
1st normal pathname entry
          . . .
          . . .
nth normal pathname entry
    1st $-option entry
          . . .
          . . .
    nth $-option entry
      default entry
```

An example of a receive print queue (in the above format) follows:

```
00005  .PATHNM2
       . . .
       . . .
00007  .PATHNM3
00003  .FILE00
       . . .
       . . .
00004  .DRL00
00001  LP01
```

Each new entry for each receive queue is placed in the queue following all entries of its type and ahead of all entries of the next type in that queue. For example, a normal pathname entry is placed in the queue following any entries of the normal variety already in the queue but ahead of any entries of the $-option variety or the default pathname. A $-option entry is placed following any normal entries and any $-option entries but ahead of the default entry. The default entry can be a $-option pathname; however, this does not affect its placement in the receive queue.

Entries in the queue are accessed in their order of appearance in the queue. The Emulator uses the entry at the head of the queue to determine 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. If the default is a $-option pathname and the value exceeds 99, the $ value is reset to 00.

When an open-pathname error is encountered, output proceeds to the next entry in the receive queue unless the error is encountered when attempting to open the default pathname. In the latter case, if the default is not a $-option file, the Emulator reattempts to open the default pathname; if the default is a $-option file, then the Emulator attempts to open $-option file 00.

To clear the receive queues, the operator can use an Abort command. ABORT*RPR, ABORT*RPU, and ABORT*ALLQ clear the receive print queue, the receive punch queue, and all queues, respectively.

The pathname specification can be of the $-option form (see paragraph 4.3.1). If this pathname form is specified, the Emulator routes data to consecutive files, beginning with <filename>00 through <filename>nn (where nn is either the maximum number of files allowed on the specified device or 99, whichever is smaller). Special SCI commands are available to delete or print all or selected $-option files. These commands, Delete Emulator Print Files (DPF) and Print Emulator Print Files (EPF), are described in paragraph 4.3.1.

When data is received, the Emulator "opens" the file designated by the first receive pathname of the appropriate queue and begins transferring data from the buffer into that file. The pathname may be that of a sequential or relative record file, a device, an IPC channel, or a spool file.

Occasionally, a reception is interrupted by the transmitting station or the receiving station through commands, physical disconnections, communication link errors, or system errors. The commands that can interrupt a reception are the Abort command and the Reset command. ABORT*TASK or ABORT*COM is issued by either the transmitting or receiving station, and transactions in progress are aborted and deleted from the queues. When a Reset command is entered at the receiving station, the transaction in progress is halted after the next received block and deleted from the queue.

A physical disconnection initiated by either station is treated as a communication link error. When a communication link error occurs, an error message is logged to the Emulator log and/or journal. Data received prior to the error is placed in the pathname specified, and the transaction is deleted from the queue. Data not received prior to the error is lost.

When a system error (usually an I/O error or an open error) occurs, the result depends on whether it occurs while the Emulator is trying to open the file specified by the receive pathname or after that file has been opened:

- If while trying to open the file specified by the receive pathname, the data goes to the next pathname in the queue. If the error occurs while trying to open the default pathname, both remote and local stations receive an error message and any data is discarded. If the error occurs while attempting to open a $-option default pathname and the pathname is not <filename>00, then the Emulator attempts to open <filename>00. If the error occurs while attempting to open the $-option pathname <filename>00, both stations receive an error message and any received data is discarded. The Emulator then reattempts to open <filename>00.

- If after opening the file specified by the receive pathname, the data received prior to the error is placed in the file, the remote station receives an error message, and the remainder of the data is discarded.

2.8 SESSION MESSAGES

The Emulator can log two types of session messages: transaction messages and statistics. Transaction messages are logged automatically in response to Emulator transaction completions. Statistics are logged in response to the Statistics (ST) command. Since transaction messages can be logged while the statistics are being logged, the messages might be noncontiguous. Also, the Emulator can optionally write operator messages to the log. Operator messages are user generated and do not necessarily describe Emulator operation. See paragraph 3.3.6.4 for information about the operator message feature.

2.8.1 Transaction Messages

The transaction messages indicate the following:

- Successful completion of transactions
- Status of transactions
- Error conditions

All messages are written to the Emulator log file and journal with the exception of one that is written to the system log (see below).

The messages are usually preceded by a carriage return and line feed; the date and time; and a four-digit number that uniquely identifies the message and its type. The message consists of a short phrase that describes the transaction, the status of the transaction, or an error condition. The templates for these messages are in the file DNCOMOBJ.TSKSRC.CCMSG on the Emulator object installation disk.

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

- A short phrase describing the error, followed by an error code and the pathname of the device or file concerned (format type 2):

  2xxx OPEN/POSITION ERROR xx pathname

- A short phrase describing the transaction, followed by the pathname of the device or file concerned (format type 3):

  3xxx RPNAME COMMAND pathname

- A short phrase describing the transaction and its status (format type 4), followed by:

  — A completion/error code

  — The transaction identification number of the transmission or the receive pathname identification of the reception

  — The number of records transferred from source to destination

  — The number of records truncated (that is, the number of records in the transfer that were longer than the maximum specified record length)

  — The pathname of the file transferred (the first pathname for a transaction with multiple pathnames) or the pathname of the device or file receiving the transmission

Example of a Session Message:

```
ddd hh:mm 4xxx RPRINT COMPLETE 00 00003 00001 00000 LP01
```

where:

ddd hh:mm	is the Julian date and time.
4xxx	is the message ID number consisting of format type and three-digit message number.
RPRINT COMPLETE	is the transaction type.
00	is the completion/error code.
00003	is the transaction ID number/receive pathname ID.
00001	is the number of records transferred.
00000	is the number of records truncated.
LP01	is the first pathname in sequence.

If the current Emulator log file or the journal is unavailable and the Emulator has a message ready for output, the following message is written to the system log:

```
** 3780/2780 WRITE ERROR TO xxxxx ERR = zz
```

where:

xxxxx is either LOG or JOURNAL.

zz is the error code. (Refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual* under internal error >00zz.)

Appendix D lists the transaction messages and the completion/error codes.

2.8.2 Statistics

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

- Current Emulator conditions — The transaction in progress, the status of the communication line, the status of features (enabled/disabled), the record and block length values, the primary and secondary command sources, the primary and secondary log destinations, the journal access name, the communication line identification, and the status of the user flags 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, IDLE, and data thresholds have been exceeded; and the terminal identification for the local and remote stations.

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

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

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

- Transactions on all queues — The transaction ID/receive pathname ID and the pathname for each transaction for each queue (presented by queue).

- Terminal ID sequence — The ID character sequence for remote and local stations.

- All statistics — A listing of all of the statistical information compiled.

The ST command and the Emulator statistics are described further in paragraph 4.5.7.

3

Functional Description

3.1 INTRODUCTION

This section describes those IBM 3780 and 2780 features and capabilities implemented by the TI DNOS 3780/2780 Emulator and those additional features that are unique to the Emulator.

3.2 EMULATED FEATURES AND CAPABILITIES

The following paragraphs describe the features and capabilities of the IBM 3780 and 2780 that are supported by the Emulator. The items discussed include the following:

- Communications support (networks supported, data link control, transmission codes, station identification)
- Operational modes (line, conversational, home)
- I/O device support (standard/optional devices, component selection, format control)
- Standard features and capabilities (retry, alarms, multiple record transmission, variable record lengths)
- Optional features (compression/expansion, end of media, transparency, binary transmissions, operator messages)

3.2.1 Communications Support

The following paragraphs describe communication lines, transmission code structure, data link control, terminal identification, and switched network control supported by the Emulator.

3.2.1.1 Communication Lines. The Emulator supports common carrier leased, switched, or equivalent, compatible, privately owned telephone facilities for its communications.

3.2.1.2 Transmission Code Structure. The IBM 3780 and 2780 terminals use Binary Synchronous Communications (BSC) procedures that support the following code sets:

- EBCDIC
- ASCII (optionally)
- Six-bit Transcode (IBM 2780 only)

The DNOS systems use the ASCII code structure internally, but the BSC procedures implemented in the DNOS 3780/2780 Emulator support EBCDIC only. ASCII and six-bit transcode transmissions are not emulated. The Emulator translates DNOS ASCII files to EBCDIC prior to transmission. When communicating with another Model 990 3780/2780 Emulator, the receiving Emulator translates the files back to ASCII after reception. Alternatively, the operator can specify nontranslation by using the Emulator's binary transmission capabilities (see paragraph 3.2.3.11). Appendix F contains ASCII/EBCDIC and EBCDIC/ASCII translation tables.

3.2.1.3 Data Link Control. The line communications that the Emulator performs follow the same BSC procedure for point-to-point exchanges used by the IBM 3780 and 2780. The Emulator does not support multipoint data link control, which enables the IBM 3780 and 2780 to operate on a communication line with multiple stations. The BSC procedure is transparent to the user and provides data verification and basic control of the data link. The data transmission is performed in half-duplex mode at line speeds up to 9600 bits per second (bps). The actual speed is determined by the user-selected modem (refer to Appendix A).

Table 3-1 lists the data link control characters that the Emulator supports and describes their most common usage. Appendix E provides additional information about the BSC procedure and data link control characters.

3.2.1.4 Terminal (Station) Identification. The Emulator can, optionally, use the terminal ID to identify its transmissions to receiving stations. This identifier is from 1 to 20 characters in length. The operator can use the ID command to specify the identifier. To display the identifier chosen for transmissions and the identifier of the remote station, if any, the operator can use the communication link Statistics command ST*ID. For further information on the ID command, refer to paragraph 4.5.

Table 3-1. Data Link Control Characters

Character	Purpose
SYN	Character provided for data synchronization. Always precedes transmitted data.
STX	Start-of-text character. Starts a transmission block.
ITB or IUS	Intermediate transmission block or information unit separator. Breaks long transmissions into optimum sizes for error checking. The 3780 Emulator can receive but not transmit an IUS.
IRS or US	Record separator or unit separator. Ends all records.
ETB	End transmission block character. Ends each block of a multiple transmission block message.
ETX	End-of-text character. Ends the last transmission block of a message.
EOT	End-of-transmission character.
ENQ	Enquiry character. The sending station has data ready to transmit.

Table 3-1. Data Link Control Characters (Continued)

Character	Purpose
ACK0 or ACK1	Acknowledgement characters. The receiving station has successfully received the transmitted data.
NAK	Negative acknowledgement character. The receiving station has received the transmitted data incorrectly, that is, with errors.
DLE	Data link escape character. Combines with other characters to provide additional control characters.
WACK	Wait and acknowledgement character. The receiving station has received the data successfully and requests that the transmitting station temporarily delay its next transmission.
RVI	Reverse interrupt character. The receiving station has received the data successfully and requests that the transmitting station stop transmitting so that the receiving station may send priority data.
TTD	Temporary transmission delay character. The sending station has data to transmit but is not quite ready and does not wish to give up the line.

3.2.1.5 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 (DISC) command from a remote station. The remote station sends a DLE EOT sequence.

- Automatically disconnect when the connection is not active. The Emulator automatically breaks the connection after a 20-second time-out if no meaningful activity has occurred. (The operator can select this feature and the time-out value during installation.) If at any time during the time-out meaningful activity takes place (such as data transfers), the time-out restarts with a new 20-second time period. The operator can use the DISC command to disconnect earlier.

- Automatically disconnect if the disconnect-on-error option was enabled during Emulator installation and any of the specified errors occur. This includes the following error situations:

  - Time-out values exceeded

  - Threshold values exceeded

— Send or receive transactions completing with error

Refer to the *Model 990 Computer DNOS 3780/2780 Emulator Object Installation Guide* for further information.

- Cancel a transmission initiated from an input device if the transmission is initiated just prior to beginning a 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 (for example, loss of carrier) from the remote station after the transmission is initiated.

The Emulator can also, optionally, send a DLE EOT character sequence (Data Link Escape, End of Transmission) to a remote station. The remote station disconnects, that is, hangs up the phone. Enabling the Disconnect Remote (DISCR) command transmits a DLE EOT to the remote station following the execution of an ABORT*TASK, ABORT*COM, DISC, or TERM command. For further information about the Disconnect Remote command, refer to Section 4.

3.2.2 Operational Modes

The online mode is the normal mode of operation for the Emulator. The following paragraphs describe the conversational and home modes of operation.

3.2.2.1 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 (Start of Text) character or a DLE STX if in transparent mode. The output device (or file) must be ready before this feature can be used. Default pathnames are assigned to the printer and the punch during installation. The operator can change these pathnames subsequent to Emulator activation by using the Receive Printer Default (RPRD) pathname and Receive Punch Default (RPUD) pathname commands, respectively. (See Appendix B and Section 4.)

3.2.2.2 Home Mode. The Emulator does not operate in home mode as do IBM terminals for 80-80 listings. DNOS utilities, not the Emulator, perform the home mode operations (for example, offline card reader to printer operations) utilizing the DNOS SCI Copy/Concatenate (CC) or Print File (PF) commands.

3.2.3 I/O Device Support

The following paragraphs describe the I/O device support features of the Emulator.

3.2.3.1 I/O Devices. The IBM 3780 and 2780 allow input from only one input device, the card reader. By contrast, the Emulator allows input from the card reader, from other input devices, from sequential or relative record files, and from IPC channels. For transmissions, the Emulator formats the data so that it appears to the remote stations to have originated 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 can direct received card punch data to any local output device, to a sequential or relative record file, or to an IPC channel. The Emulator can format data so that it can be punched at the remote station. Line printer data can be received and directed to a line printer, to any local output device, to a sequential or relative record file, to an IPC channel, or to a logical name.

Output to a device, such as a line printer, is recommended if the device is assigned a logical name. Associating a device with a logical name allows access to the device even though the DNOS spooler has access to it. Direct output to the device while the spooler is assigned to it may result in an error message containing the internal message code >009D. Refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual* for a description of this error.

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

3.2.3.2 Component Selection. The component selection feature permits the transmitting station to select the Emulator's output device(s). 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 device specified by the next available access name on the print queue.

- 3780 Emulator Component Selection. The 3780 Emulator has two selection codes for the card punch, DC2 and DC3. The operator can specify the selection code during installation. The selection code for the local station's card punch should match the card punch selection code for the remote station. The default selection code for the printer is DC1.

  The 3780 Emulator selection code is valid only when it is received as the first text characters of the first block of text, as follows:

  — The first block received after transmitting a positive response to a line bid.

  — The first block of a conversational reply.

  — The first block following a received ETX block.

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

- 2780 Emulator Component Selection. The default 2780 Emulator card punch selection code is ESC 4. The printer selection code is any of the valid EBCDIC printer ESC sequences (Table 3-2).

  The 2780 Emulator component 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, the command source must first send the selection code alone in nontransparent mode, then send the subsequent blocks as transparent data. The selection code remains in effect until another selection is made in nontransparent mode or until an EOT is sent.

In addition to receiving the selection codes, the Emulator transmits them. The selection code is automatically transmitted in the appropriate manner when the Send Data with Device Selection (SS) command is used. The command source's response to the DEVICE: prompt specifies the card punch or printer. If the selected output device is not ready, an EOT character is transmitted as a response to the block of data. The command source should not use the SS command when transmitting to an IBM host since the selection code may not be understood.

3.2.3.3 Horizontal and Vertical Format Control. The Emulator supports horizontal format control (HFC) and vertical format control (VFC) characters used by the IBM 3780 and 2780 (Table 3-2). Because certain of the printers used with DNOS systems do not use a carriage control tape, the Emulator translates the "Skip to channel n" EBCDIC VFC character to carriage return (CR) and line feed (LF). 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.

Normally, when the Emulator receives printer data that contains EBCDIC VFC or HFC characters and directs this data to a disk file, the VFC/HFC information is stripped from the file so that the standard DNOS utilities can process the data. Enabling the VFC feature allows the operator to specify that the VFC information be retained on the files in an equivalent ASCII form (refer to Table 3-2). This allows print files to be retained on a disk for later printing without sacrificing format control.

NOTE

> Print files containing carriage control characters do not have their VFC/HFC characters (CR and LF) stripped from the file when they are received into a disk file. Therefore, the operator should disable the VFC option at the receiving station to avoid adding extra VFC/HFC characters to the file. For proper printing results, print files containing carriage control characters should be transmitted with the Transparency option enabled at the transmitting station and the VFC option disabled at the receiving station.

To enable or disable the VFC option, the operator can use the Vertical Format Control (VFC) command described in Section 4 or can specify enabled or disabled as the default value during installation. The VFC option has no effect on received punch data.

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

Table 3-2. Horizontal/Vertical Format Control Characters

EBCDIC Character (and Code[1])	IBM Printer Operation	TI Printer Operation	TI ASCII Character Sequence
NL (>15)	Start a new line; terminate horizontal tab (HT); execute pending ESCs	Same	CR
IRS (>1E)	Start a new line; do not terminate HT; execute pending ESCs	Same	IRS
VT (>0B)	Vertical tab; skip to channel 2	Same	VT

Table 3-2. Horizontal/Vertical Format Control Characters (Continued)

EBCDIC Character (and Code[1])	IBM Printer Operation	TI Printer Operation	TI ASCII Character Sequence
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 / (>27 61)	Single space	Carriage return, line feed	CR LF
ESC A (>27 C1)	Skip to channel 1	Carriage return, form feed	CR FF
ESC B (>27 C2)	Skip to channel 2	Carriage return, vertical tab	CR VT
ESC C (>27 C3)	Skip to channel 3	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC D (>27 C4)	Skip to channel 4	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC E (>27 C5)	Skip to channel 5	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC F (>27 C6)	Skip to channel 6	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC G (>27 C7)	Skip to channel 7	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC H (>27 C8)	Skip to channel 8	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC I[2] (>27 C9)	Skip to channel 9	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC J[2] (>27 D1)	Skip to channel 10	Carriage return, 6 line feeds	CR LF LF LF LF LF LF

Table 3-2. Horizontal/Vertical Format Control Characters (Continued)

EBCDIC Character (and Code[1])	IBM Printer Operation	TI Printer Operation	TI ASCII Character Sequence
ESC K[2] (>27 D2)	Skip to channel 11	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC L[2] (>27 D3)	Skip to channel 12	Carriage return, 6 line feeds	CR LF LF LF LF LF LF
ESC M[2] (>27 D4)	Space suppress	Carriage return	CR
ESC S (>27 E2)	Double space	Carriage return, 2 line feeds	CR LF LF
ESC T (>27 E3)	Triple space	Carriage return, 3 line feeds	CR LF LF LF

Notes:

[1] The symbol > preceding these codes indicates hexadecimal values.

[2] The 2780 Emulator does not support this character.

3.2.3.4 Extended Enquiry (ENQ) Retry. The extended ENQ retry feature permits the IBM 3780 and 2780 to retry an ENQ 15 times before timing out with an error condition. The Emulator default retry number is also 15. The operator may increase or decrease this number during Emulator installation (see Appendix B).

3.2.3.5 Audible Alarm. The audible alarm feature on the IBM 3780 and 2780 alerts the operator that the terminal requires attention by emitting an audible alarm. 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 the log file. Table 3-3 indicates the situations that cause these messages.

Table 3-3. Emulated Audible Alarm Messages

IBM 3780 and 2780 Audible Alarm Conditions	DNOS 3780/2780 Emulator Messages
At a transmitting terminal:	
The local station does not receive a bid accepted response after an attempt to transmit.	1003 BID ERROR
An incorrect response to a redundancy check aborts the transmission.	4003 SEND COMPLETE 56 (data)[1]
A card reader or buffer error aborts the transmission.	2001 I/O ERROR (data)[1] or 4003 SEND COMPLETE 51 (data)[1]
An EOT character is received following the transmission of a block of data. This indicates that the receiving station is aborting transmission.	4003 SEND COMPLETE 51 (data)[1]
A disconnect takes place while operating in auto-answer mode on a switched line:	
— Disconnect takes place while data is being transferred.	4003 SEND COMPLETE 18 (data)[1] 1015 LINE DISCONNECTED
— Disconnect takes place while no data is being transferred.	1015 LINE DISCONNECTED
The transmission is successful.	4003 SEND COMPLETE 00 (data)[1]
At a receiving terminal:	
A printer or buffer error aborts the transmission. (An EOT transmitted in response to an abort condition.)	2001 I/O ERROR (data)[1] 4001 RPRINT COMPLETE 51 (data)[1] or 4002 RPUNCH COMPLETE 51 (data)[1]
The BEL character is received.	1002 BELL RECEIVED
No errors were detected in the transmission but records were received that exceeded the receive length (RPRRL or RPURL).[2]	4001 RPRINT COMPLETE 00 (data)[1] or 4002 RPUNCH COMPLETE 00 (data)[1]
A disconnect takes place while operating in auto-answer mode on a switched line:	

Table 3-3. Emulated Audible Alarm Messages (Continued)

IBM 3780 and 2780 Audible Alarm Conditions	DNOS 3780/2780 Emulator Messages
— Disconnect takes place while data is being transferred.	4001 RPRINT COMPLETE 18 (data)[1] or 4002 RPUNCH COMPLETE 18 (data)[1]
— Disconnect takes place while no data is being transferred.	1015 LINE DISCONNECTED
A component that is not ready is selected.	2001 I/O ERROR (data)[1]

Notes:

[1] Only the portion of these messages significant to this table is shown. For the complete message and an explanation of the item (data), refer to Appendix D.

[2] This condition is shown by 00 in the completion code field and the number of records received exceeding the receive record length (RPRRL or RPURL) in the ooooo field (see Appendix D).

3.2.3.6 Record-Length Considerations. The Emulator can transmit and receive variable-length records in certain operating environments. When deciding on the lengths to specify for the Send Record Length (SRL) command, the Receive Punch Record Length (RPURL) command, and the Receive Print Record Length (RPRRL) command, consider the following:

- The remote station. Some do not support end-of-media (EM) or compression/expansion (CE).

- The remote station's output capabilities. The send record length should be less than or equal to the remote station's output maximum.

- The local station's capabilities. The receive record lengths (punch or print) specified should be greater than or equal to the record lengths expected.

- Trailing blanks. For nontransparent transmissions, trailing blanks on the input record are not transmitted. The trailing blank count is kept in nontransparent mode only if the CE feature is enabled.

- Fixed-length records. For transmission/reception of fixed-length records, both stations should agree on the record length. To ensure that fixed-length records contain only the data intended for the record, the operator should use the Trailing Blanks (TRAILB) command to specify that the Emulator retain trailing blanks in records. This is particularly useful in receptions of relative record files. For further information about the TRAILB command, refer to Section 4.

Table 3-4 gives record-length information for several different Emulator environments. The combinations of local and remote station types included in the table are as follows:

- 990/2780 at both local and remote stations
- 990/3780 at both local and remote stations
- 990/2780 at the local station; IBM 2780 at the remote station
- 990/3780 at the local station; IBM 3780 at the remote station
- 990/2780 at the local station; IBM host computer at the remote station
- 990/3780 at the local station; IBM host computer at the remote station

The sending and receiving record lengths are specified for the following options:

- EBCDIC transparency OFF, end-of-media (EM) ON and OFF
- EBCDIC transparency ON, EM ON and OFF
- Transparency ON, binary data transmission, and EM ON and OFF (applicable in 990-to-990 communications only)

Table 3-4. Sending/Receiving Variable-Length Records

990/2780 to 990/2780

Transparency OFF

Sending records with EM ON:
The records can be of variable lengths with a maximum record length equal to the send record length (SRL).

Sending Records with EM OFF:
The records are a fixed length equal to the SRL.

Receiving records with EM ON or OFF:
The records can be of variable lengths, with a maximum equal to the receive print record length (RPRRL) for the printer or the receive punch record length (RPURL) for the punch.

Transparency ON

Sending records with EM ON or OFF:
The records are of a fixed length equal to the SRL.

Receiving records with EM ON or OFF:
The records are a fixed length equal to the RPRRL for the printer or the RPURL for the punch.

Transparency ON — Binary

Sending records with EM ON or OFF:
The records can be of variable lengths in single record blocks, with a maximum equal to the SRL. The actual record length is equal to the record length read from the input device.

Table 3-4. Sending/Receiving Variable-Length Records (Continued)

Receiving Records with EM ON or OFF:
The records may be of variable lengths in single record blocks with a maximum record length equal to the RPRRL for the printer and the RPURL for the punch. The actual record length is equal to the length of the records received.

990/3780 to 990/3780

Transparency OFF

Sending records:
The records can be of variable lengths, with a maximum equal to the SRL.

Receiving records:
The records can be of variable lengths, with a maximum equal to the RPRRL for the printer and the RPURL for the punch.

Transparency ON

Sending records:
The records are a fixed length equal to the SRL.

Receiving records:
The records are a fixed length equal to the RPRRL for the printer and the RPURL for the punch.

Transparency ON — Binary

Sending records:
The records can be of variable lengths in single record blocks. The maximum record length is equal to the SRL. The actual record length equals the record length read from the input device.

Receiving records:
The records can be of variable lengths in single record blocks. The maximum record length is equal to the RPRRL for the printer and the RPURL for the punch. The actual record length equals the record length received.

990/2780 to IBM 2780

Transparency OFF

Sending records with EM ON:
The records can be of variable lengths, with a maximum equal to either 80, 120, or 144 characters (depending on remote station requirements) when sent to an IBM printer, to 80 characters when sent to an IBM punch, or to the SRL for maximum lengths less than the device maximum.

Sending records with EM OFF:
The records are a fixed length equal to either 80, 120, or 144 characters when sent to an IBM printer, to 80 characters when sent to an IBM punch, or to the SRL for lengths less than the device maximum.

Receiving records with EM ON:
The records can be of variable lengths, with a maximum equal to 80 characters, or to the RPRRL (for the printer) or the RPURL (for the punch) for maximum lengths less than 80 characters.

Receiving records with EM OFF:
The records are a fixed length equal to 80 characters or to the RPRRL (for the printer) or the RPURL (for the punch) for lengths less than 80 characters.

Table 3-4. Sending/Receiving Variable-Length Records (Continued)

Transparency ON

Sending records with EM ON or OFF:

The records are a fixed length equal to either 80, 120, or 144 characters for the printer, to 80 characters for the punch, or to the SRL for lengths less than the device maximum.

Receiving records with EM ON or OFF:

The records are a fixed length equal to 80 characters.

990/3780 to IBM 3780

Transparency OFF

Sending records:

The records can be of variable lengths with a maximum record length equal to either 80, 120, or 144 characters (depending on remote station requirements) for an IBM printer, 80 characters for an IBM punch, or to the SRL for maximum lengths less than the device maximum.

Receiving records:

The records can be of variable lengths, with a maximum equal to 80 characters, or the RPRRL (for the printer) or the RPURL (for the punch) for maximum lengths less than 80 characters.

Transparency ON

Sending records:

The records are a fixed length equal to either 80, 120, or 144 characters for the printer, to 80 characters for the punch, or to the SRL for lengths less than the device maximum.

Receiving records:

The records are a fixed length equal to 80 characters, or to the RPRRL (for the printer) or the RPURL (for the punch) for lengths less than 80 characters.

990/2780 to IBM Host

Transparency OFF

Sending records with EM ON:

The records can be of variable lengths, with a maximum equal to 80 characters or to the SRL for maximum lengths less than 80 characters.

Sending records with EM OFF:

The records are a fixed length equal to 80 characters or to the SRL for lengths less than 80 characters.

Receiving records with EM ON or OFF:

The records can be of variable lengths, with a maximum as follows: for the printer, either 132 characters or the RPRRL (for a maximum length less than 132 characters); for the punch, either 80 characters or the RPURL (for a maximum length less than 80 characters).

Table 3-4. Sending/Receiving Variable-Length Records (Continued)

Transparency ON

Sending records with EM ON or OFF:
The records are a fixed length equal to 80 characters.

Receiving records with EM ON or OFF:
The records are a fixed length, as follows: for the printer, either 132 characters or the RPRRL (for lengths less than 132 characters); for the punch, either 80 characters or the RPURL (for lengths less than 80 characters).

990/3780 to IBM Host

Transparency OFF

Sending records:
The records can be of variable lengths with a maximum equal to 80 characters or to the SRL for maximum lengths less than 80 characters.

Receiving records:
The records can be of variable lengths, with a maximum as follows: for the printer, either 132 characters or the RPRRL (for a maximum length less than 132 characters); for the punch, either 80 characters or the RPURL (for a maximum length less than 80 characters).

Transparency ON

Sending records:
The records are a fixed length equal to 80 characters or to the SRL for lengths less than 80 characters.

Receiving records:
The records are a fixed length, as follows: for the printer, either 132 characters or the RPRRL (for lengths less than 132 characters); for the punch, either 80 characters or the RPURL (for lengths less than 80 characters).

3.2.3.7 Multiple Record Transmission. For the 2780 Emulator, the operator can specify the number of records placed in a block for transmission or reception. When communicating with an IBM host or with an IBM 2780 that has the multiple record transmission feature installed, a maximum of 7 records (or generally not more than 400 characters) may be transmitted or received. The 3780 Emulator allows as many records as can fit in a block to be transmitted or received. When communicating with an IBM 3780 or with an IBM host, as many records as can fit in a block (normally as many as 512 characters) may be transmitted or received.

3.2.3.8 Space Compression/Expansion (CE) (3780 Emulator Users Only). This feature enables the Emulator to remove strings of spaces (from 2 to 63) from data prior to transmission. The spaces are replaced by an IGS character followed by a space-count character. The IGS character indicates that spaces have been removed; the space-count character specifies the number of spaces removed. The spaces are reinstated upon reception and the IGS character is deleted. For each additional group of space characters (at least two), an additional IGS character and space-count character are inserted. This feature improves transmission efficiency and is always implemented on the receiving side in nontransparent reception. *This feature should be used in an Emulator-to-host system environment only* when the host system supports CE. It is necessary to check with the operations personnel at the host site to determine if the host system supports the CE feature. To enable or disable this feature, the operator must use the Compression/Expansion (CE) command (Section 4) or specify enabled or disabled as the default value during installation. When CE is disabled, trailing blanks are not retained on transmitted records. To retain trailing blanks, refer to paragraph 4.5.

3.2.3.9 End-of-Media (EM) (2780 Emulator Users Only). The EM feature allows the 2780 Emulator to place an end-of-record indicator at the end of the data in transmitted records. When an IBM 2780 or another emulator that supports EM reads the character, further reading of the record is prevented. The 2780 Emulator uses EM to truncate trailing blanks from input records upon transmission. This allows faster transmission of data since trailing blanks are not transmitted. To retain the trailing blanks at the end of records, refer to paragraph 4.5. To enable or disable the EM feature, the operator must use the End-of-Media (EM) command described in Section 4 or specify enabled or disabled as the default value during installation.

3.2.3.10 EBCDIC Transparency. Transparent data transmission treats data link control characters as data, using the technique known as DLE insertion. On activation of transparency, the receiving station receives a DLE STX sequence. While transparency is enabled, DLE characters precede all data link control characters, but not data characters. This allows the receiving station to identify control characters without restricting the values of data characters. The IBM 3780 and 2780 can transmit or receive all 256 eight-bit combinations in transparent mode, including binary data. However, the Emulator usually transmits and receives values in the range >00 through >7F due to ASCII-to-EBCDIC translation that masks the eighth bit. Using the binary feature allows the Emulator to transmit all 256 eight-bit combinations. However, transparency should be used to transmit data containing control characters to stations other than 990 systems. Note that transparency cannot be used with the space compression or trailing blanks feature enabled.

The characters DLE STX signal transparent transmission to the receiving station. These characters are automatically inserted when transparency is enabled using the Transparency (TY) command (Section 4) or when transparency is enabled as the default condition during installation. The characters are removed upon reception of the file at the receiving station.

The IBM 3780 and 2780 can transmit or receive all 256 eight-bit combinations, including binary data, in transparent mode. For the Emulator, transparency enabled means that the entire seven-bit ASCII code set can be transmitted or received. To transmit or receive binary data (or all 256 eight-bit combinations) with the Emulator, the command string must include the special identifier # (pound sign) immediately before the file pathname or text string. This special identifier indicates to the Emulator that it should not translate the data from ASCII to EBCDIC. Refer to the following paragraph for further information.

3.2.3.11 Binary Data Transmissions/Receptions. Binary data transmission/reception is a unique Emulator feature for use in 990-to-990 communications. It provides data transparency (DLE insertion) and inhibits ASCII-to-EBCDIC translation. In binary mode, the Emulator can transmit and receive all 256 eight-bit combinations (>00->FF). It can send and receive binary data with the Send Data (S), Send Data with Device Selection (SS), Receive Printer Data (RPR), and Receive Punch Data (RPU) commands. (For more information on these commands, refer to Section 4.)

Binary is specified using a # (pound sign). For transmissions, the pound sign precedes the first pathname that is to be sent as a binary file. All succeeding pathnames of the send command are then sent as binary files. For receptions, the pound sign precedes each pathname queued as a receive pathname for binary data. For further information about sending and receiving binary files, refer to paragraphs 4.5.3 and 4.5.4, respectively.

3.3 ADDITIONAL EMULATOR FEATURES

In addition to the IBM features and capabilities that the Emulator supports, the Emulator provides the following features:

- International character set support
- Command file operation
- IPC operation
- Local bid task feature
- Automatic call unit (ACU)
- Remote station specification features

3.3.1 International Character Set Support

The Emulator supports ASCII-to-EBCDIC and EBCDIC-to-ASCII translation for the United States, the United Kingdom, Austria, Germany, France, Belgium, Sweden, Norway, Denmark, Finland, and Japan. The country code supplied by the operating system determines which character set is used.

3.3.2 Command File Operation

Either a command file or a device using a command file can control the Emulator. A command file is a user-created sequential or relative record file that contains the Emulator commands necessary to operate the Emulator. The command file can be placed into operation during activation or at some time during the operation of the Emulator. The command file operates the Emulator in place of or in conjunction with an interactive operator. Several Emulator commands are available to give the command file many of the decision-making capabilities of the interactive operator. For further information about command files, refer to paragraphs 2.2 and 4.4.

3.3.3 IPC Operation

A task associated with an IPC channel on the DNOS system can operate the Emulator. The task must be programmed to place the commands necessary to operate the Emulator into the IPC channel queue. For further information about IPC channel operation, refer to paragraphs 2.2 and 4.4.

3.3.4 Local Bid Task Feature

The local station can bid a task for execution at the local station by using the BID command (see paragraph 4.5.7.4). This feature is normally used in the command file mode. The specification includes the following:

- The name of the program to be executed. This program must be contained in the DNOS program file .S$SHARED.
- The identification of the interactive terminal at which the task is to execute. If the default is used, the task executes independently of any terminal.
- A specification as to whether the Emulator should continue executing or be suspended during the execution of the bid task.
- Optionally, the parameters that the specified task requires.

Two bid task features are included with the Emulator. This one enables the local station to bid a specified local-station task; the other enables the local station to request that the remote station execute a specified task (see paragraph 3.3.6.3).

3.3.5 Automatic Call Unit (ACU) Feature

The ACU is a printed circuit board that can be installed on the 990 system. The ACU interfaces with the modem to allow automatic telephone dialing without operator intervention. With an ACU installed, the command source (operator or file) can issue a Call (CALL) command; the Emulator then prompts for a telephone number, which the command source supplies. The Emulator then automatically attempts to establish the connection.

Two types of ACUs are available for use 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 ACU Interface (EACUI; see Appendix A).

These two ACUs operate alike except for tandem dialing. Tandem dialing allows the dialing of an intermediate number to access an outside line (get a second dial tone) to make the telephone call. Tandem dialing is supported only for the internal ACU.

3.3.6 Remote Station Specification Features

In a 990-to-990 communication environment, the local station can send commands to the remote station by embedding them in a text string or in a record transmitted by a Send Data (S) or a Send Data with Device Selection (SS) command. This paragraph describes the conditions under which the remote station executes these commands. The following paragraphs describe the features individually. These features are specified by a modifiable four-character recognition code as follows:

- Remote receive pathname ($$$$)
- Send request ($$$S)
- Remote bid task ($BID)
- Operator message ($OPR)

Each of these features is optional and may be enabled or disabled at installation (see Appendix D) or during Emulator operation by using the appropriate command (paragraph 4.5.1). A transmitting station can send a remote station command regardless of whether that feature is enabled under its own Emulator. The receiving station recognizes only those commands that are enabled on its own Emulator.

Each of these specification features is assigned a four-character recognition code during installation. The user may either accept the default values or redefine these values to any other four-character ASCII combination. Before using any of these features, the operator at the transmitting station should determine the codes that are installed at the receiving station.

Data is transmitted in text strings or as file records. The operator can send a number of files and/or text strings by using a single S or SS command. The Emulator appends these file records and text strings into one sequential file. The Emulator receiving this file cannot distinquish records generated with text strings from records copied from a source file; both are processed identically.

When the Emulator receives a transmitted file, it examines the first four characters of the first record. If it finds any of the four-character recognition codes assigned to an enabled feature, it performs the specified action and examines the next record for further embedded commands. If it does not find a match in the first record, the Emulator treats that record and all other records of the file as data and does not examine subsequent records for embedded commands. The Emulator can recognize and act upon a command within any record as long as each previous record contains a valid and enabled command. Once the Emulator encounters a record that does not contain an executable command, no further records are examined for embedded commands.

The following demonstrates three different cases of the reception of these records:

```
$$$$.DIR.FILE*
$$$SS*O*E*80*.DIR.FILE1*
$BID*RERUN1*5*E*JAN5*
$OPR-Will the receiving station honor this specification?
```

Case 1:

If the Emulator at the receiving station has all of these features enabled and their default recognition codes installed, it performs all four specifications.

Case 2:

If the Emulator at the receiving station has $$$$, $$$S, and $OPR enabled but $BID disabled, it performs the receive pathname specification ($$$$) and the send request specification ($$$S). Since the remote bid is disabled, the Emulator does not perform the $BID specification. Since the operator message specification ($OPR) follows an unexecutable specification, the Emulator also disregards it.

Case 3:

If the Emulator at the receiving station has $$$S, $BID, and $OPR enabled but $$$$ disabled, it performs none of these specifications.

3.3.6.1 Remote Receive Pathname Feature. *In a 990-to-990 communication environment*, the transmitting station can specify the destination pathname for a transmitted file using the remote receive pathname feature. A four-character recognition code, assigned during installation, specifies to the receiving station that the pathname following the recognition code is the receive pathname for the next file transmitted. The recognition code and the destination pathname may be transmitted as one of the following:

- The first text string
- The first record of a data file
- Text strings which immediately follow an executable remote station specification
- Records of a data file which immediately follow an executable remote station specification (see paragraph 3.3.6)

For information about the operation of the remote receive pathname feature, refer to paragraphs 4.5.1.6 and 4.5.4.3.

3.3.6.2 Send Request Feature. *In a 990-to-990 communication environment*, the local station can send a request to the remote station requesting that a specified file be sent to the local station. This capability is the send request feature. The local station transmits 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 specified file to the local station. A four-character recognition code is assigned during installation to signal the receiving station that the data following the code is an S or SS command to be executed. The send request specification includes the following:

- The four-character recognition code
- A send command (S or SS)
- A character indicating transparency ON or OFF
- A character indicating EM (2780) or CE (3780) ON or OFF
- The send record length
- A destination device (if SS is the send command)
- The pathname of the file requested or the pathname of the file requested followed by the record number at which transmission should begin

The send request specification may be transmitted as one of the following:

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

- Text strings which immediately follow an executable remote station specification
- Records of a data file which immediately follow an executable remote station specification (see paragraph 3.3.6)

For information about the operation of the send request feature, refer to paragraphs 4.5.1.14 and 4.5.3.4.

3.3.6.3 Remote Bid Task Feature. *In a 990-to-990 communication environment*, the local station can send a request to the remote station requesting that it execute a specified task. The local station sends a remote bid task specification to the remote station. This specification includes the following:

- A four-character recognition code that identifies the specification as a remote bid task
- The name of the program to be executed
- The identification number of the interactive terminal at which the task is to execute
- A letter code (S or E) to specify whether the Emulator should continue executing or be suspended during the execution of the bid task
- Optionally, the parameters that the task is to use during execution

When execution of the task begins, the remote station logs a message giving the task's status.

The remote bid task specification can be transmitted as one of the following:

- The first text string
- The first record of a data file
- Text strings which immediately follow an executable remote station specification
- Records of a data file which immediately follow an executable remote station specification (see paragraph 3.3.6)

For further information on the operation of the remote bid task feature, refer to paragraphs 4.5.1.1 and 4.5.3.5.

3.3.6.4 Operator Message Feature. *In a 990-to-990 communication environment*, the local station can send operator messages to the log and journal of the remote station by using the operator message feature. The content of an operator message is strictly user defined. The Emulator recognizes an operator message as a special message and routes it accordingly, without examining its content. A four-character recognition code, assigned during installation, specifies to the remote station that the string following the recognition code is an operator message. The recognition code and message may be transmitted as one of the following:

- The first text string
- The first record of a data file
- Text strings that immediately follow an executable remote station specification
- Records of a data file that immediately follow an executable remote station specification (see paragraph 3.3.6)

When this feature is enabled, the Emulator intercepts operator messages and outputs them to the log (and journal, if specified) instead of to the designated receive pathname. For more information about the operator message feature, refer to paragraphs 4.5.1.17 and 4.5.3.6.

4

Emulator Operation

4.1 INTRODUCTION

At the local station a communication session is controlled by Emulator operations that begin with Emulator activation and end when the Emulator task completes. During the session the operator can execute only Emulator commands; the operator cannot execute SCI (System Command Interpreter) commands during this period.

The main phases of Emulator installation, activation, and operation are as follows:

- Installation. At this time, all standard and default values are assigned to Emulator parameters.
- Activation. The desired Emulator task is executed by the appropriate SCI command.
- Operation. The main phases of operation are as follows:
  - Default change operations. At this time the operator can change certain default values assigned at installation.
  - Connect operations. A telephone connection is established to the desired remote station.
  - Send/Receive operations. Bidirectional exchanges of data and control information take place.
  - Disconnect operations. The previously established telephone connection is broken.
- Deactivation. The Emulator task is terminated.

The Emulator can operate in three modes:

- Operator mode. An operator controls the session via an interactive terminal.
- Command file mode. A file (or files) controls the session; the file(s) contains a command sequence designed to control data transmission and reception in a predetermined manner.
- Interprocess communication (IPC) mode. A DNOS task that is specified as the command source controls the session.

The following additional Emulator facilities are provided:

- Status display commands. These are available only while the Emulator is active (see paragraph 4.5.7) and provide statistics and other information pertinent to the session.

- Command file commands. These commands assist in the structuring of command files and provide such facilities as conditional operation and initialization. These are available only while the Emulator is active (see paragraph 4.5.5).

- Print/Delete Emulator files. These SCI commands are available only while the Emulator is inactive and can be used only for $-option files (see paragraph 4.3.1.3).

The following conventions are used in this section when describing command prompts and responses:

- When a choice of responses is presented, the choices are enclosed in braces { }.

- User-entered responses are depicted in **BOLDFACE** type. Initial values supplied by the system are depicted in normal type.

4.2 EMULATOR ACTIVATION

The Emulator is activated from an interactive terminal by entering the appropriate SCI command Execute 3780 Emulator (X3780) or Execute 2780 Emulator (X2780). The system responds to the command by displaying the selected prompt set at the terminal as follows:

For the 2780 Emulator:

```
[ ] X2780

EXECUTE 2780 EMULATOR
            COMMAND ACCESS NAME:  ME
                LOG ACCESS NAME:  ME
            JOURNAL ACCESS NAME:
               COMM DEVICE NAME:
```

For the 3780 Emulator:

```
[ ] X3780

EXECUTE 3780 EMULATOR
            COMMAND ACCESS NAME:  ME
                LOG ACCESS NAME:  ME
            JOURNAL ACCESS NAME:
               COMM DEVICE NAME:
```

Valid responses to these menu prompts are as follows:

COMMAND ACCESS NAME:

The Emulator accepts as a valid response to this prompt the access name of any input device, file, or IPC channel allowed by the operating system (including synonyms).

The response determines the primary command source. The system displays an initial value of ME for this prompt. If the operator accepts this value, the primary command source is the operator's terminal (that is, the terminal displaying the menu).

If the operator mode is required, the operator can specify the access name of any interactive terminal or keyboard device in response to this prompt. (Device names are assigned during system generation.)

If the command file mode is required, the primary command source can be the pathname (or synonym) for a sequential file, relative record file, or IPC channel that has been previously created.

Use the Command File (CF) command to change the command source once the Emulator has been activated.

LOG ACCESS NAME:

The response to this prompt determines the primary log destination. The response is optional. The operator can enter the access name (or synonym) of any sequential or relative record file, existing IPC channel, or output device in response to this prompt. The Emulator automatically creates a sequential file for the pathname specified if it does not already exist.

If the operator requests the command file mode, and responds to the log access name prompt with a NULL value (by using erase field or skip to delete the initial value ME), no log is maintained.

If the operator has requested the operator mode, the log is automatically displayed at the interactive terminal. A second log is maintained in the operator mode if a file, device, or IPC channel name is entered in response to this prompt.

NOTE

If the operator requests the operator mode by entering the name of a command device (for example, ST04 or ME) and no other log is required, the operator should accept the default in response to the LOG ACCESS NAME: prompt. If the response to this prompt is the same as the response to the COMMAND ACCESS NAME: prompt, then only one display appears. If the command source changes the name of the command access name during the session, the log display continues; however, command input now comes from the newly specified source.

Use the Log File (LF) command to change the log destination after the Emulator has been activated.

CAUTION

When specifying a device in response to the LOG ACCESS NAME: prompt, ensure that this response does not conflict with any receive default pathname already specified at Emulator installation (either by default or user entry). A conflict causes data received from a remote station to be interspersed with the log records during communication sessions.

JOURNAL ACCESS NAME:

The response to this prompt can be any access name allowed by the operating system, such as a device name, a file pathname, an IPC channel pathname, or a synonym assigned to one of these access names. The response is optional. The Emulator accepts a null value. Since the journal is intended to be a record of Emulator activity that can be used for accounting or analysis purposes, the journal should be kept in a sequential file.

The Emulator automatically creates a sequential file with the forced write option for the pathname specified if it does not already exist. If the operator accepts the default (NULL) in response to this prompt (by pressing the RETURN key) no journal is kept. The operator cannot change the destination of the journal after the Emulator is activated.

NOTE

If the operator requested the operator mode by entering the name of a command device (for example, ST04 or ME) and no journal is required, the operator should accept the default in response to the JOURNAL ACCESS NAME: prompt. If the operator enters the same response to this prompt as to the COMMAND ACCESS NAME: prompt, only one display appears. If the command source changes the command access name during the session, the journal display continues; however, command input now comes from the newly specified source.

CAUTION

When specifying a device in response to the JOURNAL ACCESS NAME: prompt, ensure that this response does not conflict with any receive default pathname already specified at Emulator installation. A conflict causes data received from a remote station to be interspersed with the journal records during communication sessions.

COMM DEVICE NAME:

The response to this prompt is the identification of the communication line to be used. The device name has the form CMnn, where nn is the number (01 through 32) assigned to the line during system generation. Subsequent executions of the Emulator display the initial identification specified for the COMM DEVICE NAME: prompt. If the device name is correct, press either RETURN or ENTER to avoid rekeying.

The following are examples of completed prompt sets for DNOS 3780/2780 Emulators:

```
EXECUTE 2780 EMULATOR
           COMMAND ACCESS NAME:  CR01
               LOG ACCESS NAME:  ST01
           JOURNAL ACCESS NAME:  .JOURNAL
              COMM DEVICE NAME:  CM01

EXECUTE 3780 EMULATOR
           COMMAND ACCESS NAME:  ME
               LOG ACCESS NAME:  .EMLOG
           JOURNAL ACCESS NAME:  .EMJOURNL
              COMM DEVICE NAME:  CM01
```

If the user makes an error when responding to a prompt, one of the following messages appears at the activating terminal:

```
1004 COMM DEVICE INOPERABLE

1005 COMM DEVICE NOT AVAILABLE

1007 COMM COMMAND DEVICE NOT AVAILABLE

1014 INVALID PARAMETERS

1016 COMM LOG DEVICE NOT AVAILABLE

1025 COMM JOURNAL FILE NOT AVAILABLE

1026 INVALID JOURNAL/LOG ACCESS NAMES

5003 COMM DEVICE NOT AVAILABLE xx
```

Appendix D contains further details about these error messages.

If all responses are correct and if a log or journal file has been defined, the Emulator is activated and responds in the journal and all log destinations with a header display of the following form:

```
ddd hh:mm — LOG FILE xxxx ASSIGNED — RID= aa — COMM= CMbb

ddd hh:mm DNOS/xxxx v.r.e. yy.eee
```

where:

ddd is the Julian date of the execution.

hh is the hour of the execution.

mm is the minute of the execution.

xxxx is 2780 or 3780, whichever is being executed.

aa is the run ID assigned.

bb is the number of the communications device (CM) assigned.

v is the version number.

r is the release number.

e is the update level.

yy are the last two numbers of the year of the Emulator release.

eee is the Julian date of the year of the Emulator release.

The Emulator COMMAND: prompt follows the header, as shown in the following example:

```
235 13:04 — LOG FILE 2780 ASSIGNED — RID= 24 — COMM= CM01

235 13:04 DNOS/2780 1.0.0 81.110

COMMAND:
```

If the operator responds to the LOG ACCESS NAME: prompt with a null value, the first line of this display is omitted.

4.3 USING EMULATOR COMMANDS

When the Emulator task is executed, the Emulator command set becomes available at the command terminal. The SCI command set is unavailable at the command terminal until the Emulator task terminates. The operation of the Emulator consists of entering appropriate commands and command parameters in response to prompts issued by the Emulator. The following paragraphs describe the prompts and their usage. The Emulator commands are described in paragraph 4.5.

4.3.1 Command Prompts

The Emulator has 11 prompts available for display. The COMMAND: prompt is always issued to begin a command sequence and requires the entry of the selected command. Ten other prompts (DEVICE, PATHNAME, LENGTH, NUMBER, OPTION, SEQUENCE, PROGRAM, STATION, EXECUTE, and PARAMETERS) are displayed as necessary to solicit the parameters needed to execute the selected command. A valid response must follow each parameter prompt for the command to execute correctly.

An ampersand (&) response to any of the prompts aborts the command and causes the Emulator to issue a new COMMAND: prompt. A question mark (?) response results in a list of all valid responses to the prompt. This latter response to a prompt that does not have specific choices results in the message *** INVALID INPUT.

Comments included with responses to COMMAND: prompts, require at least one space between each response and the beginning of the comment.

4.3.1.1 COMMAND: Prompt. The COMMAND: prompt is the first prompt issued following activation of the Emulator or when a new command sequence begins. This prompt requires a valid Emulator command as its response. If the operator enters a command incorrectly (or the command source includes a command that is incorrect) the message *** INVALID INPUT is logged and the COMMAND: prompt reappears. Misspelled commands or commands containing embedded spaces are examples of incorrect commands.

Entering a question mark (?) in response to the COMMAND: prompt displays the legal responses to this prompt, as follows:

```
S SS RPR RPU RPRD RPUD TY CE SRL RPRRL RPURL BL RPN CALL ANS
DISCR DISC TERM RESET ABORT WAIT CF LF SRQ TBID $OPR TRAILB ST
IF $ELSE $END VFC SET CLR ID BID ?
```

The response to the COMMAND: prompt determines which other prompts are to be displayed. Each of these requires a valid response. When a command sequence is complete, a new COMMAND: prompt is displayed.

4.3.1.2 DEVICE: Prompt. The DEVICE: prompt requires an Emulator output device identification as its response. This can be either PU (for the card punch) or PR (for the printer).

4.3.1.3 PATHNAME: Prompt. The PATHNAME: prompt requires a pathname or synonym, or, for send commands, a pathname, synonym, or text string as its response. A text string may include commands directed to the remote station (see paragraph 3.3.6).

Pathname Responses. A pathname may be the name of a device, file, IPC channel, or logical name. If a pathname represents a device, the pathname is the name assigned to that device during system generation (for example, LP01, DS01). The device name can be from one to four characters in length. The first character must be alphabetic; the rest can be alphabetic or numeric.

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; periods separate components of the pathname. The maximum length of a pathname is 48 characters. The volume name, directory name, and file name can each be one to eight characters in length; the first character must be alphabetic and the rest can be alphabetic or numeric. The volume name need not be included if the volume is the system disk. A file name can be concatenated directly to a volume name without requiring a directory.

IPC channels are identified by pathnames similar in structure to file pathnames.

The following are examples of valid DNOS pathnames:

Pathname	Description
VOLN.DIR.FILE	The pathname of a file named FILE, located in directory DIR, located on volume VOLN.
DEVN.DIR.FILE	The pathname of a file named FILE, located in directory DIR, located on device DEVN.
.DIR.FILE	The pathname of a file named FILE, located in directory DIR, located on the system volume.
.FILE	The pathname of a file named FILE, located on the system disk.
VOLN.FILE	The pathname of a file named FILE, located on the volume VOLN.

Synonym Responses. A string of one or more characters (synonym) can substitute for a pathname and is a valid response to the PATHNAME: prompt. A synonym can represent an entire pathname or a component of a pathname. The Assign Synonym (AS) SCI command allows synonym assignments. The following example shows a synonym assigned to the file VOL.DIRB.DIRC.FILE:

```
[ ] AS

ASSIGN SYNONYM VALUE
          SYNONYM:  ABCD
            VALUE:  VOL.DIRB.DIRC.FILE
```

The following lists some possible synonyms:

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

$-Option Pathnames. The Receive Printer Data (RPR), Receive Printer Default Pathname (RPRD), Receive Punch Data (RPU), and Receive Punch Default Pathname (RPUD) commands may use a special form of pathname, called $-option. Placing $ at the end of any standard pathname converts it to a $-option pathname. The character $ is a default value assigned during Emulator installation. Alternatively, the pound sign (#) instead of $ can be specified during installation (see Appendix B). In the following discussion, all references to $ apply to # where # has been specified instead of $.

CAUTION

If the $ default applies, ensure that any pathname used as a $-option pathname does not contain another $ sign. During processing, the first $ sign encountered is interpreted as a file name termination character. If it is necessary to process files whose names contain $, specify the # sign during installation.

Optionally, during Emulator installation, $-option default receive pathnames may be specified instead of standard pathnames for the RPR and RPU commands.

During the operation of the Emulator, $ is assigned a consecutive number in the range 00 through 99 (or some value less than 99 if the capacity of the device or directory is exceeded) each time a file is received on the queue containing the $-option pathname. The file name or directory name appended with $ must not exceed its maximum length (see "Pathname Responses" above) when the $ sign is replaced by this two-digit number; that is, the directory or file must not exceed 46 characters excluding $ which is the 47th character. If a file already exists with a pathname that is to be used by the $-option, that file is written over with the $-option file.

Pathname		**$-Option**
Standard	**$-Option**	**Description**
VOLN.DIR.FILE	VOLN.DIR.FILE$	Consecutively specifies files FILE00 through FILEnn in directory DIR on volume VOLN.
DEVN.FILE	DEVN.FILE$	Consecutively specifies files FILE00 through FILEnn on device DEVN.

The Emulator software package includes two special SCI commands that print and delete $-option files after the Emulator creates them. These commands are Print Emulator Print Files (EPF) and Delete Emulator Print Files (DPF). They cannot execute during Emulator operation. The commands print or delete all or selected $-option files created during an Emulator communication session.

EPF Command. To execute EPF, enter the DNOS SCI command mode and then enter EPF following the SCI command prompt, [ ]. The following prompt set appears:

```
[ ] EPF

PRINT EMULATOR PRINT FILES
      OUTPUT DEVICE NAME:  LP01
          BASE PATHNAME:  .PRINT
        STARTING OFFSET:  00
          ENDING OFFSET:  99
                 DELETE:  NO
  NUMBER OF LINES/PAGE:  62
            ANSI FORMAT:  NO
                   FORM:  STANDARD
          BANNER SHEET?:  YES
```

The status of the print queue is displayed after the operator responds to the above prompts and the specified files are queued. Table 4-1 describes the responses required. If the Emulator encounters an undefined file or error while executing the EPF command, printing proceeds to the next file.

DPF Command. To execute DPF, the operator must enter the DNOS SCI command mode and then enter DPF following the SCI command prompt, [ ]. The following prompt set appears:

```
[ ] DPF

DELETE EMULATOR PRINT FILES
        BASE PATHNAME:  .PRINT
      STARTING OFFSET:  00
        ENDING OFFSET:  99
```

Table 4-2 describes the responses required. If the Emulator encounters an undefined file or error while executing the DPF command, execution of the command terminates immediately, and the appropriate error message is displayed.

Table 4-1. EPF Command Prompts and Responses

System Prompts	User Responses
OUTPUT DEVICE NAME:	The device name of the printer that is to print the files. Required response. The default is LP01.
BASE PATHNAME:	The pathname to which 00 through nn is appended to create the $-option files. Required response. The default is .PRINT.
STARTING OFFSET:	The beginning number of a range of file numbers or the specific number of the file to be printed. Required response. The default is 00.

Table 4-1. EPF Command Prompts and Responses (Continued)

System Prompts	User Responses
ENDING OFFSET:	The ending number of a range of file numbers or the specific number of the file to be printed. Required response. The default is 99.
DELETE:	Respond with YES to delete the file(s) after printing. Optional response. The default is NO.
NUMBER OF LINES/PAGE:	Enter the integer value for the number of lines to be printed per page. The default is 62 lines. Optional response.
ANSI FORMAT:	Respond with YES if the file contains American National Standards Institute (ANSI) control characters. Optional response. The default is NO.
FORM:	Enter the name of the form that should be mounted on the output device. The default form is STANDARD. Refer to the Print File (PF) command in the *DNOS SCI Reference Manual* for more information. Optional response.
BANNER SHEET?:	Respond with YES if an identifier sheet is to be printed as the first page of output preceding only the first file of the range of files to be printed. Optional response. The default is YES.

Table 4-2. DPF Command Prompts and Responses

System Prompts	User Responses
BASE PATHNAME:	The base pathname is the pathname to which 00 through nn was appended to create the $-option files. Required response. The default is .PRINT.
STARTING OFFSET:	The beginning number of a range of file numbers or the specific number of the file to be deleted. Required response. The default is 00.
ENDING OFFSET:	The ending number of a range of file numbers or the specific number of the file to be deleted. Required response. The default is 99.

4.3.1.4 LENGTH: Prompt. The LENGTH: prompt requires a record or block length specification (in bytes) as its response. Appendix B provides possible value ranges.

4.3.1.5 NUMBER: Prompt. The NUMBER: prompt requires a telephone number, a records-per-block specification, or a flag number as its response. The telephone number format is detailed in the description of the CALL command (paragraph 4.5.2.2). Appendix B provides the value range for the number of records per block. Paragraph 4.5.5.1 describes the numbering system for the flags.

4.3.1.6 OPTION: Prompt. The OPTION: prompt requires a valid option as its response. The valid options are listed in the discussion of each command.

4.3.1.7 SEQUENCE: Prompt. The SEQUENCE: prompt requires a terminal identifier that is from 1 to 20 characters long. Paragraph 4.5.7.3 describes the characters that may be used in the identifier.

4.3.1.8 PROGRAM: Prompt. The PROGRAM: prompt requires a valid program name as its response. The program must be listed in the DNOS program file .S$SHARED.

4.3.1.9 STATION: Prompt. The STATION: prompt requires a valid interactive terminal identification as its response (for example, ST04) or a synonym (for example, ME).

4.3.1.10 EXECUTE: Prompt. The EXECUTE: prompt requires a YES or NO as its response. The resulting action is described with the command using this prompt.

4.3.1.11 PARAMETERS: Prompt. The PARAMETERS: prompt requires up to four valid ASCII characters as its response. The significance of these characters is described with the command using this prompt.

4.3.2 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:. Additional prompts vary according to the command. Commands can be issued in general or condensed format. Usually, the general format is used only in the operator mode; however, the condensed format may be used with advantage irrespective of the type of command source.

4.3.2.1 General Format. The general format begins with the COMMAND: prompt. The response is a valid command followed by a carriage return. The next prompt for this command, if any, is then issued. Each prompt issued requires an appropriate response followed by a RETURN. To initiate execution of the command, do one of the following, as appropriate:

- Enter a valid response for the final prompt and press RETURN.
- Follow the next to final response with an asterisk and press RETURN.
- Respond to the final prompt by pressing RETURN (null response).

When the primary command source is other than an operator, each record (line) is considered a complete response. Examples of the general format for activating a command are as follows:

EXAMPLES

```
COMMAND:   RPN
OPTION:    ON

COMMAND:   S
PATHNAME:  .DIR.FILE*

COMMAND:   SS
DEVICE:    PR
PATHNAME:  .DIR.FILE
PATHNAME:  .DIR.FILE1
PATHNAME:  return
```

4.3.2.2 Condensed Format. The condensed format also begins with a response to the COMMAND: prompt. A condensed entry requires that the command entered in response to the prompt be followed by its parameters on the same line. Asterisks are required to delimit the command and each of the parameters.

The following examples are the same commands as given in the previous paragraph, but in the condensed format:

```
COMMAND:   RPN*ON*

COMMAND:   S*.DIR.FILE*

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

Within a command file, prompts supplied by the Emulator, such as COMMAND: are omitted. Omitting the asterisk as a termination to a response when the command still requires additional parameters causes the Emulator to issue the appropriate prompt for the next expected parameter; that is, the command returns to the general format for the remainder of the command or until the command source enters a parameter followed by an asterisk. If the omission is by mistake, the operator can correct the situation by pressing RETURN or entering an asterisk, as appropriate to the particular command.

For the command file mode and the IPC mode, the condensed format is more convenient to use than the general format. 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. Appendix C contains examples of command files using the condensed format.

4.3.2.3 Use of the Terminating Asterisk. Omission of the terminating asterisk is allowed when there are no more input parameters, as in the Statistics command (for example, either ST*ALL* or ST*ALL is acceptable). However, the terminating asterisk is mandatory with 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. Also, the terminating asterisk is mandatory with the condensed format of the CALL command when specifying manual dial on a system configured with an ACU. In this case, the terminating asterisk follows the command mnemonic CALL.

Examples of the Use of the Terminating Asterisk:

An example of the condensed format terminated with an asterisk (indicating that .DIR.FILE1 is the last pathname) is as follows:

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

The following is an example of the condensed format without a terminating asterisk; the asterisk is not necessary because there are no more input parameters:

```
COMMAND:  RPRRL*100
```

An example of the condensed format with the terminating asterisk omitted in error is as follows; the condensed format returns to the general format with its next prompt:

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

In the following example, the system is equipped with an ACU, but an asterisk is used to terminate the CALL command because the operator wishes to dial the call manually. The asterisk prevents display of the NUMBER: prompt:

```
COMMAND:  CALL*
```

In the next example, no terminating asterisk is required since the system is not equipped with an ACU (that is, no other parameters are allowed). After issuing the CALL command, the operator manually dials the call; or, alternatively, if the communication line is a leased line, a connection is automatically attempted:

```
COMMAND:  CALL
```

4.3.2.4 Use of the Ampersand. Including an ampersand (&) in a command string aborts that command. Another COMMAND: prompt then appears. However, an ampersand enclosed in angle brackets (< >) is accepted in the response to the PATHNAME: prompt of an S or SS command. Normally, in the operator mode, an operator does not use an ampersand in the condensed format since an incorrect response to the COMMAND: prompt can be corrected by backspacing and retyping the response. However, the condensed format is included for comparison purposes in the examples on the following pages.

Examples of the Use of the Ampersand:

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

```
          General Format                  Condensed Format

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

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

```
          General Format                  Condensed Format

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

COMMAND:   next command           COMMAND:   next command
```

In the next example, the ampersand is part of a text string enclosed in angle brackets and does not abort the command:

```
          General Format                  Condensed Format

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

The following is an example of the ampersand entered outside the text string, thus aborting the command:

```
          General Format                  Condensed Format

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

4.3.2.5 Use of the Question Mark. Entering a question mark (?) response to a prompt causes a list of all valid responses to that prompt to be displayed and the prompt to be repeated. A question mark 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.

Examples of the Use of the Question Mark:

In the following example, the operator enters ? to determine all possible responses to the OPTION: prompt of the RPN command:

```
       General Format                      Condensed Format

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

In the next example, the operator uses ? to determine all possible responses to the OPTION: prompt of the ABORT command:

```
       General Format                      Condensed Format

    COMMAND:  ABORT                     COMMAND:  ABORT*?*
    OPTION:   ?                         COM TASK ALLQ RPR RPU S ?
    COM TASK ALLQ RPR RPU S ?           ANY VALID TRANSACTION ID
    ANY VALID TRANSACTION ID            OPTION:   option
    OPTION:   option
```

4.3.2.6 Use of the Comma. A comma embedded in a response to a PATHNAME: prompt is interpreted as a pathname separator. The comma can appear within pathname responses to the S or SS command, as shown in the following examples.

Examples of the Use of the Comma:

The following example sends the files .DIR.PAGE1 and .DIR.PAGE2:

```
       General Format                      Condensed Format

COMMAND:   S                        COMMAND:  S*.DIR.PAGE1,.DIR.PAGE2*
PATHNAME:  .DIR.PAGE1,.DIR.PAGE2
PATHNAME:  return
```

The next example sends the file .DIR.PAGE1 and the text string <THIS IS A TEST> with the card punch selected as the destination device:

```
       General Format

COMMAND:   SS
DEVICE:    PU
PATHNAME:  .DIR.PAGE1,<THIS IS A TEST>
PATHNAME:  return

       Condensed Format

COMMAND:   SS*PU*.DIR.PAGE1,<THIS IS A TEST>*
```

4.3.2.7 Use of the Blank. A blank embedded in a response is interpreted as a terminator. This feature allows the inclusion of comments 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.

Examples of the Use of the Blank:

The following general format 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
```

The following example shows blanks within a text string that is enclosed in angle brackets:

```
        General Format                     Condensed Format

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

In the following example, a blank is entered in error. The intended command is 'RPRRL'; 'RPRR' is not a valid command. The Emulator outputs an error message and prompts for another command.

```
        General Format                     Condensed Format

COMMAND:   RPRR L                  COMMAND:   RPRR L*100*
*** INVALID INPUT                  *** INVALID INPUT

COMMAND:   next command            COMMAND:   next command
```

4.4 MODES OF OPERATION

The Emulator can operate in three modes:

- Operator mode
- Command file mode
- Interprocess communication (IPC) mode

The mode of operation determines the specific commands required and the organization of the communications session.

The following paragraphs describe the modes of operation. The Emulator command set is described in paragraph 4.5 and summarized in Appendix G.

4.4.1 Operator Mode

In the operator mode, the operator issues commands, responds to prompts, and responds to Emulator messages at a keyboard device. The operator chooses the commands to be executed according to the desired operation of the Emulator. Emulator commands can be entered in general or condensed format.

The log destination is automatically the operator's terminal. The operator can direct an additional log to a relative record or sequential file, an output device, or an IPC channel by using the LF command or by designating a log file during activation.

In operator mode, the operator can use the CF command to specify a command file or another keyboard device as a command source. The command file is a predefined set of commands and responses to command prompts established on a relative record or sequential file. Once specified, by using the CF command, the command file becomes the secondary command source for the session. Control returns to the operator when any of the following occurs:

- The last command of the command file does not terminate the task or specify another command file or IPC channel.
- A second command file or IPC channel is specified with a null pathname (that is, CF*).
- An error occurs in reading commands from the command file.

4.4.2 Command File Mode

In the command file mode, a file containing the Emulator commands required to achieve a desired sequence of operations is stored in a specified sequential or relative record file. Each record in the command file can be a set of one or more responses to a single command prompt. Although the format of the response can be either general or condensed, condensed is more convenient. Neither format should include the prompts to which the data in the file responds, such as: COMMAND:, PATHNAME:, or DEVICE:. In the general format, each response is one record; null responses are represented by blank records. Each response has a maximum record length of 80 characters. See Appendix C for examples of command files.

The commands and command parameters in the command file are organized according to the desired operation of the Emulator. Since the operator is not able to interact with the command file while it is operating, the command file must be set up to include all commands necessary to accomplish the interaction and to react to any problems that may occur. The use of the IF and WAIT commands enables the command file to make decisions based on the operation of the Emulator.

The log destination can be the operator's terminal, a sequential or relative record file, an output device, or an IPC channel. Execution of the Show File (SF) SCI command (with appropriate responses) causes display of the log file while the Emulator is writing to it.

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 or $END is encountered in the command stream. When $ELSE is encountered, all commands following the $ELSE are skipped until $END is encountered. If the specified condition is false, the Emulator skips all commands until $ELSE or $END is encountered. $ELSE or $END is a valid response to a COMMAND: prompt.

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 Emulator can check for the following conditions when the IF command is used:

- Connection
- Disconnection
- Send queue empty
- Receive print queue empty
- Receive punch queue empty
- All queues empty
- Send completed
- Receive punch completed
- Receive print completed
- Most recent transaction completed
- User flag set
- 2780 Emulator
- 3780 Emulator

The IF*EM2780 and IF*EM3780 Emulator conditions allow the command file to check which Emulator, 3780 or 2780, is activated before a set of commands is executed. This is useful in systems that have both a 2780 Emulator and a 3780 Emulator installed.

The user flag set condition is used to conditionally execute or skip a group of commands if a flag is set. During installation, the maximum number of user flags that may be set and cleared during Emulator operation can be specified. The flags are usually set conditionally, as shown in the following example:

```
IF*SS*
SET*1*
$ELSE
S*.DIR.FILE*
$END
```

After the flags are set, they can be checked conditionally to cause a specified action if the flag is not set (that is, if the condition is false). The flags are set to zero initially (when the Emulator is activated) and can be conditionally cleared.

An example of an application that might use the flags in this way is an environment in which a local station must send one data file to two different remote stations. Command files designed to accomplish this are illustrated in Appendix C.

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 waits for a connection, a disconnection, a queue to become empty, or all queues to become empty. The WAIT command has safeguards built into it to prevent the command file from waiting indefinitely for a condition that is never to be satisfied. These safeguards are as follows:

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

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

After a command file has executed, no interruption of the operation of the Emulator is allowed unless the task is killed or the connection is broken. For this reason, ensure the following when arranging the command file:

- That all directories of a multilevel pathname are available. If they are not, ensure that a default pathname is specified.

- That the number of transactions queued prior to issuing a CALL command does not exceed the number specified in response to the PENDING REC/SEND REQUESTS: prompt during installation. If the number waiting exceeds the number specified, the Emulator waits for queue buffer space to be available; however, without a connection or a connect in progress, the last valid command issued is aborted and the command prompt is reissued.

A command file can be either the primary or the secondary command source. The command file operation is almost the same in either case. The major difference is that a primary command source command file can issue a CF command to execute a second command file without closing, whereas a secondary command source command file automatically closes after it completes execution.

After a secondary command source file opens, it issues its commands and then returns control to the primary command source, or (if it issued a CF command) to a new command file. When the secondary command source command file issues a CF command to open a new command file, the secondary command source command file closes and the newly specified command file begins execution. When the command file completes execution, control returns to the point in the primary command source from which the call to the first secondary command source was invoked, and not to the first secondary command source.

When the command file is the primary command source, it controls the session until one of the following occurs:

- An end-of-file (EOF) is read.
- A TERM command is encountered.
- An ABORT*TASK command is encountered.
- An error occurs during the command sequence and automatic attempts to recover are unsuccessful.

When any of these occurs, the Emulator task terminates.

When the command file is the secondary command source, it controls the session until one of the above conditions occurs 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 the secondary command source is designated as a VDT, then an error or EOF (generated by pressing the ENTER key) terminates the Emulator task. If a CF command is encountered in the current command file, the executing command file is terminated and the new one begins execution. Issuing a CF command with a null (blank) pathname closes the command file that is currently executing and returns session control to the primary command source at the command following the CF command that invoked the secondary command file.

4.4.3 IPC Mode

Since DNOS recognizes IPC channels as I/O devices and accesses them by pathnames, a DNOS task is a valid command source or log destination. The commands issued by the DNOS tasks are records that are identical to command file records. The commands are written to and read from individual symmetric channels by the DNOS task. These channels are defined as symmetric, job-local channels at creation. A symmetric channel is one in which the owner and requestor tasks issue simple read and write commands in a one-to-one correspondence. The reads of one task are processed as soon as the other task issues a write command and vice versa.

Each IPC channel is defined as an exclusive channel. Multiple tasks cannot access one Emulator with one channel.

The Emulator executes the commands in the same way that it executes a command file written to a relative record or sequential file. The DNOS task can add to the queue as it executes. In this way, the task can control the operation of the Emulator to some extent. Still, since an operator cannot interact with the Emulator while it is executing queued commands, use of the Emulator requires careful planning in advance; the IF and WAIT commands execute in the same way that they do in a command file, and the same care that is taken in planning a command file must be taken in planning the commands that are placed on an IPC channel.

Using an IPC channel as the primary command source does not require using an IPC channel for the log destination. The Emulator can accept commands from an IPC channel and output log information to a terminal, a relative record or sequential file, a device, or another IPC channel. However, to input sufficient information to the DNOS task, log file analysis may be necessary.

EXAMPLE

As an example of the IPC mode of operation, consider a user-written COBOL task that controls Emulator operations in an unattended mode. The COBOL task is arranged to write properly formatted Emulator commands to an IPC channel with a pathname, say Y, in response to data read from another pathname, say Z. The COBOL task reads Emulator-generated messages and prompts from Z. The two IPC channels, Y and Z, must be created with the appropriate characteristics (symmetric type, job-local scope, nonshared access, and message length 80). The COBOL task is the owner of both channels and issues reads and writes to these channels, using the designated pathnames.

The Emulator is now activated with Y and Z specified as the command and log access names, respectively. As the last prompt response is completed, the COBOL task is automatically activated and assumes complete control of Emulator operations. In this example, the COBOL task serves as a sink for Emulator log output and also as a source for Emulator commands.

For further information about IPC and the creation and operation of IPC channels, refer to the *Model 990 Computer DNOS Concepts and Facilities* manual.

4.5 EMULATOR COMMANDS

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

- Change default
- Connect
- Send data
- Receive data
- Command file
- Disconnect
- Miscellaneous

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), and receive print and punch default pathnames, as well as for enabling or disabling optional features. The change default commands allow changes in the default values as necessary during the session. Appendix B lists the installed defaults.

Execution of the ST command (see paragraph 4.5.7.2) displays the current values or status of installed defaults as follows:

- ST*ALLQ — Shows the current receive print and receive punch default pathnames.
- ST*CUR — Shows the status or value of all other modifiable defaults.

4.5.1.1 Remote Bid Task Command (TBID). Use the TBID command to enable or disable the remote bid task feature. This feature enables a receiving station to recognize and process a request from a transmitting station to execute a specified task. The remote bid task specification is identified by a four-character recognition code established at installation time.

General Format	*Condensed Format*
COMMAND: TBID OPTION: $\left\{\begin{array}{l}\text{ON}\\\text{OFF}\\\text{?}\end{array}\right\}$	COMMAND: TBID* $\left\{\begin{array}{l}\text{ON}\\\text{OFF}\\\text{?}\end{array}\right\}$ *

EXAMPLES

General Format	*Condensed Format*
COMMAND: **TBID** OPTION: **ON**	COMMAND: **TBID*ON***
COMMAND: **TBID** OPTION: **OFF**	COMMAND: **TBID*OFF***

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

General Format	*Condensed Format*
COMMAND: BL LENGTH: block length (bytes)	COMMAND: BL*block length*

In 990-to-990 communications, the transmit block length entered must exceed the send record length by at least three bytes to allow the inclusion of protocol characters. Block lengths specified as less than the receive or transmit record lengths are considered invalid and result in the return of the message *** INVALID INPUT.

The block length for the Emulators can be any decimal or hexadecimal number in the decimal range 80 through 1024 (>50 through >400). A hexadecimal number must be preceded by either the sign > or a zero (0). However, the installation default values for the 3780 and the 2780 Emulators are not the same (see Appendix B).

For communications with an IBM host system, the block length should be 400 (>190) bytes for the 2780, or 512 (>200) bytes for the 3780.

NOTE

> Block lengths outside the limits specified may be valid when communicating with an IBM host system. However, consider transmission efficiency when specifying these lengths.

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**BL**	COMMAND:	**BL*288***
LENGTH:	**288**		

4.5.1.3 Compression/Expansion Command (CE) (3780 Emulator Users Only). Use the CE command to enable or disable the compression/expansion feature. This feature removes consecutive space characters from input data before transmission and restores them on reception.

General Format		*Condensed Format*	
COMMAND:	CE	COMMAND:	CE*$\begin{Bmatrix}\text{ON}\\ \text{OFF}\\ \text{?}\end{Bmatrix}$*
OPTION:	$\begin{Bmatrix}\text{ON}\\ \text{OFF}\\ \text{?}\end{Bmatrix}$		

If transparency is on (TY*ON), the Emulator ignores the CE command.

Some host systems do not support compression and expansion. Keep this in mind when specifying ON or OFF for this feature in the 3780 Emulator.

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**CE**	COMMAND:	**CE*ON***
OPTION:	**ON**		
COMMAND:	**CE**	COMMAND:	**CE*OFF***
OPTION:	**OFF**		

Specifying CE ON or OFF affects transmitted data only; the Emulator automatically processes received data correctly, whether or not the remote station uses the compression/expansion feature.

4.5.1.4 Disconnect Remote Command (DISCR). The DISCR command causes a DLE EOT transmission to the remote station when an ABORT*TASK, ABORT*COM, DISC, or TERM command is executed. The DLE EOT sequence causes the remote station to disconnect.

General Format	*Condensed Format*
COMMAND: DISCR	COMMAND: DISCR*$\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}$*
OPTION: $\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}$	

The communication line must be a switched line for this command to execute successfully. Also, a connection must have been previously established with the remote station for the ABORT*COM, ABORT*TASK, DISC, or TERM commands to disconnect the remote station.

EXAMPLES

General Format	*Condensed Format*
COMMAND: **DISCR** OPTION: **ON**	COMMAND: **DISCR*ON***
COMMAND: **DISCR** OPTION: **OFF**	COMMAND: **DISCR*OFF***

4.5.1.5 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 receiving station reads the indicator, no further reading of the record occurs. Trailing blanks are truncated, thereby improving transmission efficiency. The end-of-media feature is valid for nontransparent data only.

General Format	*Condensed Format*
COMMAND: EM	COMMAND: EM*$\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}$*
OPTION: $\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}$	

Some host systems do not support the end-of-media feature; therefore, specify EM ON or OFF for the 2780 Emulator according to the host system in use.

Specifying EM ON or OFF affects transmitted data only; the Emulator automatically processes received data correctly, whether or not the end-of-media character is present. If the transparency feature is enabled (TY*ON), the EM command is ignored.

To restore trailing blanks on data record reception use the optional TRAILB command.

EXAMPLES

```
General Format                 Condensed Format

COMMAND:  EM                   COMMAND:  EM*ON*
OPTION:   ON

COMMAND:  EM                   COMMAND:  EM*OFF*
OPTION:   OFF
```

4.5.1.6 Remote Receive Pathname Command (RPN). Use the RPN command to enable or disable the remote receive pathname feature. With RPN enabled, the Emulator looks for the four-character recognition code (established during installation) of the remote receive pathname, followed by a pathname in the received data. With RPN disabled, the Emulator treats the four-character recognition code and pathname as data.

General Format / *Condensed Format*

COMMAND: RPN — COMMAND: RPN*$\left\{\begin{array}{l}\text{ON}\\ \text{OFF}\\ ?\end{array}\right\}$*

OPTION: $\left\{\begin{array}{l}\text{ON}\\ \text{OFF}\\ ?\end{array}\right\}$

The remote receive pathname recognition code established during installation at the receiving station must match the code sent by the transmitting station. Otherwise, the received characters are processed as data.

EXAMPLES

```
General Format                 Condensed Format

COMMAND:  RPN                  COMMAND:  RPN*ON*
OPTION:   ON

COMMAND:  RPN                  COMMAND:  RPN*OFF*
OPTION:   OFF
```

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

General Format		*Condensed Format*	
COMMAND:	RPRRL	COMMAND:	RPRRL*record length*
LENGTH:	record length (bytes)		

The record length can be any decimal or hexadecimal number from one to the maximum output buffer size specified during installation. A hexadecimal number must be preceded by either a > sign or a zero. Otherwise it is interpreted as decimal.

Received records that are longer than the specified record length are truncated to the specified length. The number of records truncated is included in the reception complete message on the log and/or journal.

When specifying the receive print record length, consider the following:

- Local station printer characteristics (for example, buffer size and paper width)
- The length of the records that the remote station transmits

To avoid truncation, specify a receive print record length equal to or greater than the record lengths being received. For transmission/reception of fixed-length records, both stations should agree on the record length.

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**RPRRL**	COMMAND:	**RPRRL*132***
LENGTH:	**132**		

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

General Format		*Condensed Format*	
COMMAND:	RPRD	COMMAND:	RPRD*pathname*
PATHNAME:	pathname		

The $-option pathname is allowed as the default pathname. If .DIR.FILE$ were entered for the default pathname, consecutive files .FILE00 through .FILEnn would be used. Note the following:

- If an error occurs during an attempt to open one of the consecutive files (for example, .FILE04) the Emulator begins writing to .FILE00.
- If the Emulator cannot open .FILE00, an error message is logged to the log and/or journal files (see Appendix D), with the $-option count not incremented; that is, the identity .FILE00 is not incremented to .FILE01.
- If the maximum number of $-option files allowed (99) is reached, the Emulator resets the consecutive file counter and directs the next receptions to .FILE00.

The Emulator prohibits the removal of the default pathname from the receive print queue; this pathname remains unchanged until this command is used again or the Emulator is reactivated.

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**RPRD**	COMMAND:	**RPRD*.DIR.FILE***
PATHNAME:	**.DIR.FILE**		

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

General Format		*Condensed Format*	
COMMAND:	RPUD	COMMAND:	RPUD*pathname*
PATHNAME:	pathname		

The $-option pathname is allowed as the default pathname. If .DIR.FILE$ were entered for the default pathname, consecutive files .FILE00 through .FILEnn would be used. Note the following:

- If an error occurs during an attempt to open one of the consecutive files (for example, .FILE04) the Emulator begins writing to .FILE00.
- If the Emulator cannot open .FILE00, an error message is logged to the log and/or journal files (see Appendix D), with the $-option count not incremented; that is, the identity .FILE00 is not incremented to .FILE01.
- If the maximum number of $-option files allowed (99) is reached, the Emulator resets the consecutive file counter and directs the next receptions to .FILE00.

The Emulator prohibits removal of the default pathname from the receive punch queue; this pathname remains unchanged until this command is used again or the Emulator is reactivated.

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**RPUD**	COMMAND:	**RPUD*.DIR.FILE***
PATHNAME:	**.DIR.FILE**		

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

General Format		*Condensed Format*	
COMMAND:	RPURL	COMMAND:	RPURL*record length*
LENGTH:	record length		

The record length can be any decimal or hexadecimal number from one through the maximum output buffer size specified during installation. A hexadecimal number must be preceded by either a > sign or a zero.

When specifying the receive punch record length, consider the following:

- Local station card punch characteristics
- The length of the records that the remote station transmits

To avoid truncation, specify an RPURL that is equal to or greater than the record lengths being sent. For communications using fixed-length records, both stations should agree on the record length. The number of records truncated during the reception is included in the reception complete message.

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**RPURL**	COMMAND:	**RPURL*60***
LENGTH:	**60**		

4.5.1.11 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		*Condensed Format*	
COMMAND:	RB	COMMAND:	RB*number*
NUMBER:	number		

When communicating with an IBM host, a maximum of seven records per block is 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. The multiple record transmission feature allows the IBM 2780 to receive a maximum of seven records, or 400 characters per block.

EXAMPLE

```
        General Format                Condensed Format

        COMMAND:  RB                  COMMAND:  RB*5*
        NUMBER:   5
```

4.5.1.12 Retain Trailing Blanks (TRAILB). Use the TRAILB command to restore trailing blanks in received records. With the TRAILB option enabled, the Emulator fills the remainder of each received record with blanks, up to the number of bytes specified by the RPRRL or RPURL command. This enables the Emulator to receive fixed-length records when a fixed length is required for processing. When the VFC option is on (VFC*ON), the TRAILB command is disabled (TRAILB*OFF).

When the Emulator is writing received data into a relative record file, enable the TRAILB option. Relative record files are inherently fixed length; hence, using this option to fill the remaining portion of a relative record file record with blanks facilitates I/O with user-written tasks.

```
      General Format                  Condensed Format

                                                      (ON  )
   COMMAND:  TRAILB              COMMAND:  TRAILB*  { OFF  }*
                                                      (?   )

              (ON  )
   OPTION:   { OFF  }
              (?   )
```

EXAMPLE

```
        General Format                Condensed Format

     COMMAND:  TRAILB              COMMAND:  TRAILB*ON*
     OPTION:   ON
```

4.5.1.13 Send Record Length Command (SRL). Use the SRL command to specify a record length (in bytes) for transmitted data.

```
           General Format                  Condensed Format

   COMMAND:  SRL                      COMMAND:  SRL*record length*
   LENGTH:   record length (bytes)
```

The record length can be any decimal or hexadecimal number from one to the maximum input buffer size specified during installation. A hexadecimal number must be preceded by a > sign or a zero. The transmitted block length must exceed the send record length entered by at least three bytes to allow the inclusion of protocol characters.

When specifying the send record length, consider the following:

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

The send record length specified should be equal to or less than the remote station's specified receive record lengths. For fixed length record communications, both stations must agree on the record length.

Records from the specified input source that are longer than the current send record length are truncated to the specified length. The number of records truncated during a transmission is included in the send complete message.

Records from the specified input source that have trailing blanks are transmitted without those blanks. Restoration of the blanks is accomplished at the receiving station with the TRAILB option.

EXAMPLE

```
   General Format                     Condensed Format

   COMMAND:  SRL                      COMMAND:  SRL*80*
   LENGTH:   80
```

4.5.1.14 Send Request Command (SRQ). Use the SRQ command to enable or disable the send request feature. When this command is executed at the local station, a remote station may transmit a message to the local station, requesting that the local station send a specified file to the remote station. The send request is identified by a four-character recognition code established during installation. Enabling the SRQ feature at the local station enables the station to recognize the four-character recognition code and act on the send request. When the SRQ feature is disabled, the four-character recognition code is treated as data.

```
   General Format                     Condensed Format

                                                   (ON )
   COMMAND:  SRQ                      COMMAND:  SRQ*{OFF}*
                                                   (?  )
              (ON )
   OPTION:   {OFF}
              (?  )
```

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**SRQ**	COMMAND:	**SRQ*ON***
OPTION:	**ON**		
COMMAND:	**SRQ**	COMMAND:	**SRQ*OFF***
OPTION:	**OFF**		

4.5.1.15 Transparency Command (TY). Use the TY command to enable or disable the EBCDIC transparency feature for transmitted data. The transparency feature allows the transmission and reception of any binary-configured data, including the full EBCDIC and ASCII character sets.

General Format		*Condensed Format*	
COMMAND:	TY	COMMAND:	$\text{TY*}\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}\text{*}$
OPTION:	$\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ ?\end{matrix}\right\}$		

With transparency disabled (TY*OFF), only a restricted set of ASCII characters (translated to EBCDIC) can be transmitted. With transparency enabled (TY*ON), the full set of ASCII characters, including control characters, can be transmitted. Refer to Appendix F for a list of the characters that cannot be transmitted with the transparency feature disabled.

Binary data (data that is to be transmitted 'as is', unconverted to EBCDIC) can be transmitted only in the transparent mode. However, additional requirements exist for successful communications using this data format; refer to paragraphs 4.5.3 and 4.5.4 for further information.

Selection of transparency or nontransparency affects only transmitted data; the Emulator processes received data in either state correctly.

When EBCDIC transparency is enabled for the Emulator, any specification of end-of-media or space compression/expansion is ignored; that is, end-of-media and compression/expansion apply only to nontransparent data.

The Emulator can direct received transparent data to the station printer; however, any EBCDIC characters less than >40 without an ASCII equivalent are not printed; any EBCDIC characters greater than or equal to >40 without an ASCII equivalent are printed as blanks.

Transparent data must be transmitted in fixed-length records. In communications between a 990 computer and an IBM 2780, 3780, or host system, the record lengths for the receiving station's output device should determine the send record length; the record length of the remote station's input determines the receive record length (print or punch). In communications between two 990 computers, the send record length of the transmitting station determines the record length; the receive record length should be set accordingly. Refer to Table 3-4 for further information on send and receive record lengths.

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**TY**	COMMAND:	**TY*ON***
OPTION:	**ON**		
COMMAND:	**TY**	COMMAND:	**TY*OFF***
OPTION:	**OFF**		

4.5.1.16 Vertical Format Control Command (VFC). Use the VFC command to enable (ON) or disable (OFF) the VFC feature. Enabling this feature allows the Emulator to define (in ASCII) the VFC characters in a received data file output to a disk. Refer to Table 3-2 for a list of the recognized VFC characters, and their converted ASCII equivalents.

```
        General Format                      Condensed Format

                                                      ( ON  )
COMMAND:   VFC                     COMMAND:   VFC*  { OFF } *
                                                      ( ?   )

           ( ON  )
OPTION:    { OFF }
           ( ?   )
```

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**VFC**	COMMAND:	**VFC*ON***
OPTION:	**ON**		
COMMAND:	**VFC**	COMMAND:	**VFC*OFF***
OPTION:	**OFF**		

4.5.1.17 Operator Message Command ($OPR). Use the $OPR command to enable or disable the operator message feature. When this feature is enabled, the Emulator examines the first four characters of text strings and file records received, looking for the four-character recognition code of the operator message feature. When the Emulator encounters this code, it outputs the remainder of the text string or record as an operator message to the log (and journal, if specified) and not to the designated receive pathname.

The four-character recognition code for the operator message feature is established during installation. The default value for this code at installation is $OPR (see Appendix B). The recognition code that the transmitting station uses must match the code that the receiving station uses; otherwise, the code is processed as data. When the operator message feature is disabled, the recognition code and any associated message are treated as data.

General Format		*Condensed Format*
COMMAND:	\$OPR	COMMAND: $\text{\$OPR*}\begin{Bmatrix}\text{ON}\\ \text{OFF}\\ \text{?}\end{Bmatrix}\text{*}$
OPTION:	$\begin{Bmatrix}\text{ON}\\ \text{OFF}\\ \text{?}\end{Bmatrix}$	

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**$OPR**	COMMAND:	**$OPR*ON***
OPTION:	**ON**		
COMMAND:	**$OPR**	COMMAND:	**$OPR*OFF***
OPTION:	**OFF**		

4.5.2 Connect Commands

The connect commands, ANS or CALL, accept and initiate respectively a communications link connection via the specified types of network.

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

EXAMPLE

COMMAND: **ANS**

Unless ringing equipment is installed, do not use the ANS command for a leased line. Attempting to use this command with a leased line otherwise results in the nonassertion of the DTR (Data Terminal Ready) signal and failure to establish a connection.

4.5.2.2 Call Command (CALL). Use the CALL command to establish a connection by placing a telephone call in both a switched network and a leased line environment.

General Format		*Condensed Format*	
COMMAND:	CALL	COMMAND:	CALL*telephone number*
NUMBER:	telephone number		

The NUMBER: prompt is issued only if the Emulator configuration includes an automatic call unit (ACU). A null response (carriage return) to the NUMBER: prompt either specifies manual dialing (even if an ACU has been configured in the Emulator) or specifies that the communication link is a leased line where dialing is not required.

The telephone number format is the same as for normal telephones. The number consists of the decimal digits required to access the switched network and place a call to the remote station. Insert additional characters, if desired, as follows:

- Embedded dashes and parentheses to facilitate readability. The Emulator omits these when outpulsing is initiated.

- Equal sign (=) characters to indicate that a wait for a dial tone is necessary. These are not required for the dial tone that occurs immediately following the initial off-hook.

Note the following restrictions:

- Embedded blanks are not allowed in the telephone number entry.

- The asterisk (*) and pound (#) signs are not allowed. The normal functions of these signs in the AT&T and associated networks are retained in the Emulator as follows:

  — The asterisk is replaced by any ASCII character whose least significant four bits are 1010 (for example, : or Z).

  — The pound sign is replaced by any ASCII character whose least significant four bits are 1011 (for example, +).

  Note that these two characters are valid only for subscriber connections using tone dialing. Do not use them for subscriber connections normally using rotary dials.

Space for 80 characters (the maximum length allowed for an operator command input record) is available for telephone number entries. Telephone numbers required for accessing remote stations (including international points) from the AT&T and associated networks normally do not exceed 15 digits. However, privately controlled networks that also utilize the public networks may require the entry of other numerical data in addition to the telephone number.

An example of a telephone number entry is 9 = 1-(512)-327-3639. This entry signifies that a call is being made from within a PBX system and that the Emulator must wait for an intermediate dial tone before continuing to outpulse the area code, local exchange code, and line number.

Note that Western Electric (Bell) modems and ACUs do not support waits for intermediate dial tones (termed tandem dialing).

When a connection is established, the following message is displayed and written to the log and/or journal:

```
(date/time) 1010 CONNECT SUCCESSFUL
```

If an attempt to establish a connection has failed for any reason (such as busy signal or wrong number) the following message is displayed and written to the log and/or journal:

```
(date/time) 5002 CALL RETRY NECESSARY xx
```

The term xx is a system error code. For an explanation, refer to Appendix D and the *Model 990 Computer DNOS Messages and Codes Reference* manual.

The following examples show the use of the CALL command.

Examples of the Use of the CALL Command:

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

```
      General Format                Condensed Format

COMMAND:  CALL                COMMAND:  CALL*327-3639*
NUMBER:   327-3639
```

The following example shows a CALL command issued on a system equipped with an ACU when the number is dialed manually.

```
      General Format                Condensed Format

COMMAND:  CALL                COMMAND:  CALL*
NUMBER:   return
```

The following example shows a CALL command issued on a system without an ACU. No number prompt is displayed. The number is dialed manually or the communication line is a leased line that does not require dialing.

```
      General Format                Condensed Format

COMMAND:  CALL                COMMAND:  CALL*
```

4.5.3 Send Data Commands

The send data commands are used to queue data for transmission over the communication link. The following paragraphs include these topics:

- Using the Send Data and Send Data with Device Selection commands
- Using the send request feature
- Using the remote bid task feature
- Using the operator message feature

The Send Data (S) and Send Data with Device Selection (SS) commands are executed from the send queue; this is an ordered queue of all send commands in the order of their entry (first-in, first-out). Once the send command is queued, it is referred to as a transaction and is assigned a unique number called a transaction ID. The transaction ID assigned is not reassigned to another send, and therefore, can be used to follow the progress of a transaction while it is waiting for Emulator attention. This transaction ID can be found in the corresponding 'send initiated' and 'send complete' message. The Emulator does not provide for the allocation of priorities to send transactions; use selective deletion of send transactions from the queue to achieve some degree of priority (see paragraph 4.5.6.1).

In general format, a null response (return) to a PATHNAME: prompt terminates the send commands. In the condensed format, ending a line record with an asterisk (*) terminates the send commands. After a send command and all associated pathnames are correctly entered, the Emulator displays the following message:

```
068 08:01 1013 ID 00001
```

where:

068 is the Julian date.

08:01 is the time.

1013 is the Emulator message ID.

00001 is the transaction ID.

The response to a PATHNAME: prompt can be one or more pathnames or text strings separated by commas or asterisks and not exceeding 48 characters. 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 (specified during installation) available at the time the command is entered. When multiple pathnames are specified, the files are concatenated and sent as one file. Use any valid pathname in response to a PATHNAME: prompt.

A record number enclosed in parentheses can follow the pathname or synonym to designate at which record of the file the send should begin. The record numbers allowed are from 1 through 65,535. Entering a record number of zero results in the message *** INVALID INPUT. Entering a record number beyond the end of the file results in no transmission and the logging of an I/O error with an error code of 30. The record number may be decimal or hexadecimal. When a hexadecimal number is used, it must be preceded by either a zero or a > sign.

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 cannot exceed 48 characters. Any ASCII character is allowed in a text string. To abort the command with an ampersand (&) in this situation, first close the text string with an angle bracket and then use the ampersand.

When a send command is executed, the Emulator parameters for the transmitted data (such as transparency ON/OFF) are queued with the command. (The current state of these parameters is displayed using the Display Current Statistics, ST*CUR, command.) All change default commands should therefore be executed before the send command. The parameters affect only the transmitted data.

4.5.3.1 Binary Data Transmissions. For binary data transmissions (that is, data files required to be transmitted as is, unconverted to EBCDIC) in a 990-to-990 communication environment, the Emulator can initiate transmissions using the send data commands. Binary data is specified using a pound sign (#) and can be specified only once in each send command. The first pathname or text string to be sent as a binary file must be preceded by a pound sign; each succeeding pathname or text string specified with that send command is also sent as a binary file. The pound sign specifies to the Emulator that the file should not be translated before transmission.

Binary files are transmitted in single record blocks with transparency enabled. The Emulator automatically enables the transparency feature for each binary transmission; manual specification is unnecessary. The transparency option, as displayed, is not changed to the ON condition.

DNOS program files and directories must be transferred as binary files. The user must perform the following steps before file transfer:

1. Ensure that the transmitting and receiving stations both specify a record length of 256 bytes.

2. Execute a Backup Directory (BD) SCI command, specifying the pathname of the file(s) to be transferred. Enable the Block option as shown in the following example:

   ```
   [ ] BD

     BACKUP DIRECTORY
              DIRECTORY PATHNAME:  .DNOSFILE
          SEQUENTIAL ACCESS NAME:  .SEQFILE
             CONTROL ACCESS NAME:
             LISTING ACCESS NAME:  ME
                         OPTIONS:  ALIASES,BLOCK,NODATE
   ```

   Specifying the BLOCK option generates a submenu. Replace the default value (9600) with 256, as follows:

   ```
   BD:  BLOCKING FACTOR IN BYTES
           BLOCKING FACTOR:  256
   ```

3. At the transmitting station, precede the file pathname with a pound sign (#), specifying that translation is not required. Use this pathname in the S or SS command.

4. At the receiving station, precede the file pathname with a pound sign, specifying that translation is not required. Use this pathname in the RPR or RPU command.

NOTE

It is most important that the receiving station operator precede the receive pathname with a pound sign when preparing to receive files containing binary data. If this is not done, received data is assumed to be in EBCDIC and all received data is translated. No indication of the error appears at the receiving station.

The receiving DNOS system must execute a Restore Directory (RD) command before using the file. (Refer to Table 3-4 for a list of file transmission environments and the parameters affecting record length.)

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

General Format

```
COMMAND:   S
PATHNAME:  { pathname[(record number)] }
           { <text string>             }
  . . .       . . .
  . . .       . . .
  . . .       . . .
  . . .       . . .
PATHNAME:  return
```

Condensed Format

```
COMMAND:  S*{ pathname[(record number)] }*. . .*
            { <text string>             }
```

In the general format, the Emulator continues to issue PATHNAME: prompts until it receives a null response (return). In the condensed format, an asterisk (*) terminates the send command. After a send command and all associated pathnames are correctly entered, the Emulator displays the date, the time, the Emulator message ID, and the transaction ID. The record number enclosed in parentheses designates the file record at which the send should begin. It is enclosed in brackets to indicate that it is optional.

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

Examples of the Use of the Send Data (S) Command:

The following example sends the file .DIR.FILE:

```
         General Format                    Condensed Format

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

The following example sends the file .DIR.FILE beginning with record number 124:

```
        General Format                     Condensed Format

   COMMAND:   S                       COMMAND:   S*.DIR.FILE(124)*
   PATHNAME:  .DIR.FILE(124)
   PATHNAME:  return
```

The following example sends the file .DIR.FILE and text string <THIS IS TEXT>:

```
        General Format                    Condensed Format

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

The following is another example of sending the file .DIR.FILE and the text string <THIS IS TEXT>:

```
          General Format                      Condensed Format

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

The following example sends the file .DIR.FILE as binary data; transparency is automatically enabled.

```
         General Format                    Condensed Format

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

The following example sends the files .DIR.FILE and .DIR.FILE1; the files are sent as binary data with transparency automatically enabled.

```
         General Format                          Condensed Format

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

The following example sends the file .DIR.FILE1 with the destination specified as pathname .DIR.FILE.

```
         General Format                          Condensed Format

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

The following example sends the files .JCL and .BINDATA. The file .JCL is to be sent as nontransparent data, and the file .BINDATA is to be sent as binary (untranslated) data with transparency ON:

```
         General Format                          Condensed Format

COMMAND:   S                          COMMAND:  S*.JCL,#.BINDATA*
PATHNAME:  .JCL
PATHNAME:  #.BINDATA
PATHNAME:  return
```

The following example sends the request that the file .DIR.FILE be queued to be sent to the local station. The file being queued has transparency ON, compression/expansion OFF (3780 Emulator users only) or end-of-media OFF (2780 Emulator users only), a record length of 132, and the printer selected as the destination device.

```
              General Format

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

              Condensed Format

COMMAND:  S*<$$$SSS*T*O*132*PR*.DIR.FILE*>*
```

4.5.3.3 Send Data with Device Selection Command (SS). Use the SS command to enter a data file or concatenation of files 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:    {PR}
           {PU}
PATHNAME:  {pathname[(record number)]}
           {<text string>           }
  . . .       . . .
  . . .       . . .
  . . .       . . .
  . . .       . . .
PATHNAME:  return
```

Condensed Format

```
COMMAND:   SS*{PR}*{pathname[(record number)]}*. . .*
              {PU} {<text string>           }
```

In the general format, the Emulator continues to issue PATHNAME: prompts until it receives a null response (return). In the condensed format, an asterisk (*) terminates the SS command. After a send command and all associated pathnames are correctly entered, the Emulator displays the date, the time, the Emulator message ID, and the transaction ID. The record number enclosed in parentheses designates the file record at which the send should begin. It is enclosed in brackets to indicate that it is optional.

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. Because device selection is preset in IBM host systems, they do not recognize device selection for received data. Device selection for data sent to an IBM 3780 or IBM 2780 station or to another Emulator is recognized and honored.

Examples of the Use of the Send Data with Device Selection (SS) Command:

The following example sends 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*
```

The following example sends the file .DIR.FILE, beginning with record number 124, with the card punch selected as the destination device:

```
              General Format                  Condensed Format

     COMMAND:   SS                  COMMAND:  SS*PU*.DIR.FILE(124)*
     DEVICE:    PU
     PATHNAME:  .DIR.FILE(124)
     PATHNAME:  return
```

The following example sends the file .DIR.FILE and the text string <THIS IS TEXT> with the card punch selected as the destination device:

```
            General Format                  Condensed Format

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

The following example sends the file .DIR.FILE and the text string <THIS IS TEXT> with the printer selected as the destination device:

```
           General Format                   Condensed Format

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

The following example sends the files .DIR.FILE and .DIR.FILE1 with the printer selected as the destination device. Both files are sent as binary files since .DIR.FILE is preceded by a pound sign. Also, both files are sent with transparency automatically enabled.

```
            General Format                   Condensed Format

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

4.5.3.4 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 or vice versa.

When using the send request feature, the requesting station sends an augmented Send Data (S) command or an augmented Send Data with Device Selection (SS) command to the responding station for execution. The additional information consists of the following:

- A specification of transparency ON or OFF
- End-of-media (EM) ON or OFF (for 2780 Emulators)
- Compression/expansion (CE) ON or OFF (for 3780 Emulators)
- The send record length

To use this feature enter one of the following at the local station in response to the first PATHNAME: prompt of the Send Data (S) command:

- A text string consisting of the send request information. Multiple send requests are possible using consecutive records with different send request information in each record.
- The pathname of a file whose first record consists of the send request information.
- The pathname of a file whose only record consists of the send request information.

Paragraph 3.3.6 describes in detail the conditions necessary for the successful execution of remote specifications. The send request information is as follows:

- A four-character send request recognition code. This code can be the standard default recognition code $$$S 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.

NOTE

If transparency is enabled, consideration of the end-of-media and compression/expansion options is redundant since neither option is permissible in transparent mode.

- 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 pathname may be followed by the record number (enclosed in parentheses) at which transmitting should begin.

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:

RRRR is the four-character send request recognition code.

S or SS is the send command.

T is the single character transparency designation.

E is the single character end-of-media or compression/expansion designation.

RL is the one to three digit send record length.

DD is the two-character device selection (if SS is the command sent).

pathname is the pathname or synonym of the file requested.

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) values in the send request do not affect the quiescent condition of these options at the responding station; also the new states of the options are not displayed as a changed value during statistical displays. Those values contained in the send request are implemented only in connection with the execution of the request.

Upon completion of a successful transmission of the requested file by the responding station, the condition of these options returns to the previous values. The send request transaction is transparent to the responding station except for the logging of messages associated with the transaction and the presence of the S or SS command in the station's send queue. When specifying the record length in the send request, consider the maximum input buffer size of the local (requesting) station and the maximum output buffer size of the remote (responding) station honoring the send request to avoid truncation of records.

When a synonym is used as the send request pathname, the remote station operator must define the synonym for the remote system.

The remote station does not honor the send request recognition code unless the send request feature is enabled at that station. Enable the send request feature by using the send request (SRQ) command (paragraph 4.5.1.14) or enable it 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.

EXAMPLES

The following example sends the request that the file .DIR.FILE be queued for transmission to the local station. The file has transparency OFF, compression/expansion ON (3780 Emulator users only) or end-of-media ON (2780 Emulator users only), and a record length of 80.

General Format

```
COMMAND:  S
PATHNAME: <$$$SS*O*E*80*.DIR.FILE*>
PATHNAME: return
```

Condensed Format

```
COMMAND:  S*<$$$SS*O*E*80*.DIR.FILE*>*
```

The following example sends the request that the file .DIR.FILE be queued for transmission to the local station. The file has transparency OFF, compression/expansion OFF (3780 Emulator users only) or end-of-media OFF (2780 Emulator users only), and a record length of 132. The printer is selected as the destination device.

General Format

```
COMMAND:  S
PATHNAME: <$$$SSS*O*O*132*PR*.DIR.FILE*>
PATHNAME: return
```

Condensed Format

```
COMMAND:  S*<$$$SSS*O*O*132*PR*.DIR.FILE*>*
```

4.5.3.5 Using the Remote Bid Task Feature. In a 990-to-990 communication environment, the remote bid task feature can be used by a transmitting station to request that the remote station execute a specified task (compare the BID command, paragraph 4.5.7.4). The Send Data (S) or Send Data with Device Selection (SS) command is used to send a bid task specification to the remote station. The remote bid task specification may be transmitted as a text string (that is, enclosed in angle brackets), as the first record of a file, or as the only record of a file. The remote bid task specification includes the following:

- A four-character remote bid task recognition code. This code is established during installation. Use the default value, $BID, or a user-defined value.

- The program name of the program to be executed. The program name can be from one to eight characters long; the first character must be alphabetic, but the rest can be alphabetic or numeric. The program name must be installed in the program file .S$SHARED.

- The identification number of the interactive terminal at which the task executes. The terminal identification is a decimal or hexadecimal number. A hexadecimal number must be preceded by a zero. The identification number may be from one to three digits long, including the zero, if used. If a user terminal identification is used, that terminal is used by the task until the task completes or is aborted. If the terminal selected is already in use, an error is returned on the bid. Using the identification number >FF bids the task independently of a terminal. Using the terminal identification of 0 (zero) executes the task at the receiving station.

- The execution code. The letter S suspends the Emulator task when the task is bid; E continues execution of the Emulator after the task is bid.

- The calling parameters (optional). Pass user-specified parameters to the bid task. The parameters must be contained in four characters and the bid task must issue a Get Parameters supervisor call (SVC) to obtain the parameters.

The components of the remote bid task specification are separated by asterisks. Use the following format for the remote bid task specification:

```
code*progname*sid*rc*parm*
```

where:

code	is the remote bid task recognition code (default value $BID).
progname	is the program name.
sid	is the interactive terminal identification.
rc	is the execution code.
parm	is the optional calling parameters.

The receiving station must have the remote bid task feature enabled (TBID*ON) to recognize the remote bid task recognition code. Otherwise, the remote bid task specification is considered data. Paragraph 3.3.6 describes in detail the conditions necessary for successful execution of remote specifications. After the bid is received, the receiving station attempts to execute the task by using the Execute Task supervisor call (>2B). (See the *DNOS Supervisor Call (SVC) Reference Manual.*) After the attempt, the receiving station logs a message indicating successful or unsuccessful completion of the bid.

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**S**	COMMAND:	**S*<$BID*RERUN1*5*E*JAN5*>***
PATHNAME:	**<$BID*RERUN1*5*E*JAN5*>**		
PATHNAME:	return		
COMMAND:	**S**	COMMAND:	**S*<$BID*ACCTNG*0FF*S*>***
PATHNAME:	**<$BID*ACCTNG*0FF*S*>**		
PATHNAME:	return		

4.5.3.6 Using the Operator Message Feature. In a 990-to-990 communication environment, the local station can send operator messages to the log and journal of the remote station by using the operator message feature. An operator message may be specified in a text string or in the first record of any file sent by a local station using the Send Data (S) or the Send Data with Device Selection (SS) command. Paragraph 3.3.6 describes in detail the necessary conditions for the successful execution of remote specifications. During Emulator installation, the user assigns a four-character recognition code to this feature or accepts the default value, $OPR. Use the following format for the operator message specification:

CODEoperator message

where:

CODE is the four-character recognition code for the operator message feature. The default value is $OPR.

operator message is the message text, a string of up to 48 characters. The message string immediately follows the recognition code. No delimiting characters are used.

When this feature is enabled, the Emulator examines the first four characters of received data file records and text strings. When these characters match the recognition code, the Emulator treats the remainder of the string or record as an operator message. The Emulator discards the recognition code and inserts the date, the time, and the operator message ID at the beginning of the message. The Emulator writes this information to the log (and journal, if specified). The general format of the operator message output is as follows:

```
ddd hh:mm <message number> <operator message>
```

where:

ddd is the Julian day.

hh is the hour.

mm is the minute.

<message number> is a unique number identifying the type of message (see Appendix D).

<operator message> is a string from 1 to 48 characters long.

The Emulator accepts text strings in response to any PATHNAME: prompt. Text strings are enclosed in angle brackets (< >) and cannot exceed 48 characters in length. When the operator sends a message using a text string, the first four characters must be the four-character recognition code. This limits the length of the message to 44 characters.

When sending an operator message within the first record of a file, the operator can input a message longer than 48 characters. However, the remote station writes only the first 48 characters of the message to the log, ignores the rest, and reports no errors.

Examples of the Use of the Operator Message Feature:

The following example sends an operator message within a text string.

General Format

```
COMMAND:   S
PATHNAME:  <$OPRThis is an example of an operator message.>
PATHNAME:  return
```

Condensed Format

```
COMMAND:   S*<$OPRThis is an example of an operator message.>*
```

Either format results in the following output to the remote log and/or journal:

```
365 23:59 7001 This is an example of an operator message.
```

The following example sends an operator message within the first record of a data file. It is assumed that the first record of the data file, VOL.DIR.MYFILE, is the following:

```
$OPRThis is an operator message sent from MYFILE.
```

```
         General Format                      Condensed Format

COMMAND:   S                       COMMAND:   S*VOL.DIR.MYFILE*
PATHNAME:  VOL.DIR.MYFILE
PATHNAME:  return
```

Both formats produce the same result. The first record in MYFILE is interpreted as an operator message. All other records in MYFILE are sent to the remote destination device. The log entry at the remote station is as follows:

```
001 01:01 7001 This is an operator message sent from MYFILE.
```

The following example uses the Send Data with Device Selection (SS) command to send an operator message to the remote log and a text string and the file MYFILE to the file DEST.

General Format

```
COMMAND:   SS
DEVICE:    PR
PATHNAME:  <$OPRem2780-5575   *Generate new W-4s>
PATHNAME:  <$$$$.DIR.DEST>
PATHNAME:  <This is a header for the DEST file.>
PATHNAME:  VOL.DIR.MYFILE
PATHNAME:  return
```

Since this example is too long to fit into one record, it cannot be implemented with the condensed format. The following is sent to the remote station log:

```
002 08:00 7001 em2780-5575   *Generate new W-4s
```

4.5.4 Receive Commands

The receive commands specify the pathnames for received data designated for the printer or punch. These commands are entered to either the receive print or receive punch queue for execution.

Once the receive command is queued, it is referred to as a transaction and is assigned a unique number called a transaction ID. The transaction ID assigned is not reassigned to any other transaction and therefore can be used to follow the progress of a transaction while it is awaiting Emulator attention. This transaction ID can be found in the corresponding 'receive initiated' and 'receive complete' message.

Both general and condensed formats are available with these commands. In both formats, the receive commands are terminated with a response to the PATHNAME: prompt. After the command and pathname are correctly entered, the Emulator displays the date, the time, the message ID, and the transaction ID.

The pathname specification may be of the $-option form. If this pathname form is specified, received data is routed to consecutive files, beginning with <filename>00 through <filename>nn (where nn is either the maximum number of files allowed on the specified device or 99, whichever is less) and continuing to the next entry on the queue.

For binary data reception in a 990-to-990 communication environment, a pound sign must precede the receive pathname specified in the RPR or RPU command. The pound sign specifies to the Emulator that the file should not be translated to EBCDIC.

Binary data is automatically transmitted with transparency ON and in single record blocks. The transmitting and receiving stations should agree on the maximum record length.

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

DNOS program files and directories must be transferred as binary files. The user must perform the following steps before file transfer:

1. Ensure that the transmitting and receiving stations both specify a record length of 256 bytes.

2. Execute a Backup Directory (BD) SCI command, specifying the complete pathname of the file(s) to be transferred. The block option must be enabled as shown in the following example:

   ```
   [ ] BD

     BACKUP DIRECTORY
            DIRECTORY PATHNAME:  COMM.DNOSFILE
       SEQUENTIAL ACCESS NAME:  .EM3780
          CONTROL ACCESS NAME:
          LISTING ACCESS NAME:  ST01
                      OPTIONS:  NOALIASES,BLOCK,NODATE
   ```

   Specifying the BLOCK option generates a submenu. Replace the initial value (9600) with 256, as follows:

   ```
   BD:  BLOCKING FACTOR IN BYTES
          BLOCKING FACTOR:  256
   ```

3. At the transmitting station, precede the file pathname with a pound sign, specifying that translation is not required. Use this pathname in the S or SS command.

4. At the receiving station, precede the file pathname with a pound sign, specifying that binary translation is not required. Use this pathname in the RPR or RPU command.

NOTE

> It is most important that the receiving station operator precede the receive pathname with a pound sign when preparing to receive files containing binary data. If this is not done, received data is assumed to be in EBCDIC and all received data is translated. No indication of the error appears at the receiving station.

The receiving DNOS system must execute a Restore Directory (RD) SCI command before using the program file(s) within the directory. (Refer to Table 3-4 for a list of file transmission environments and the parameters affecting record length.)

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

General Format	*Condensed Format*
COMMAND: RPR	COMMAND: RPR*pathname*
PATHNAME: pathname	

The VFC characters (Table 3-2) 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 ASR/KSR terminal, the output is single spaced. The VFC characters are translated to format files directed to the line printer.

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

General Format	*Condensed Format*
COMMAND: RPU	COMMAND: RPU*pathname*
PATHNAME: pathname	

Examples of the Use of the Receive Printer Data (RPR) and Receive Punch Data (RPU) Commands:

The following example specifies receive print file in pathname .DIR.FILE:

General Format	*Condensed Format*
COMMAND: RPR	COMMAND: RPR*.DIR.FILE*
PATHNAME: .DIR.FILE	

The following example specifies receive binary print file in pathname .DIR.FILE:

General Format	*Condensed Format*
COMMAND: RPR	COMMAND: RPR*#.DIR.FILE*
PATHNAME: #.DIR.FILE	

The following example specifies receive consecutive print files in pathnames .DIR.FILE00 through .DIR.FILEnn:

General Format	*Condensed Format*
COMMAND: RPR	COMMAND: RPR*.DIR.FILE$*
PATHNAME: .DIR.FILE$	

The following example specifies receive punch file in pathname .DIR.FILE:

```
          General Format                  Condensed Format

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

The following example specifies receive binary punch file in pathname .DIR.FILE:

```
          General Format                  Condensed Format

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

The following example specifies receive consecutive punch files in pathnames .DIR.FILE00 through .DIR.FILEnn:

```
          General Format                  Condensed Format

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

4.5.4.3 Using the Remote Receive Pathname Feature. When the remote receive pathname feature is enabled at the receiving station, the transmitting station can specify the remote receive pathname for a transmitted file.

To use this feature, enter 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 included as one of the following:

- The pathname or synonym of a file whose first record consists of the remote receive pathname recognition code followed by the destination pathname.

- The pathname or synonym of a file whose only record consists of the remote receive pathname recognition code followed by the destination pathname.

- A text string consisting of the remote receive pathname recognition code followed by the destination pathname.

The format of the remote receive pathname is as follows:

$$\text{nnnn} \left\{ \begin{matrix} \text{pathname} \\ \text{synonym} \end{matrix} \right\}$$

where:

nnnn is the recognition code. This code is either the standard default recognition code $$$$ or a user-defined recognition code set during Emulator installation.

The receiving station will not recognize the remote receive pathname recognition code unless the remote receive pathname feature is enabled at the receiving station. To enable this feature, either use the Remote Receive Pathname (RPN) command (paragraph 4.5.1.6) or enable the feature during installation. If the feature is not enabled and the station receives a remote receive pathname designation, the station treats the designation as data. Paragraph 3.3.6 describes in detail the necessary conditions for the successful execution of remote specifications.

With the remote receive pathname feature enabled, all files and text strings following the receive pathname specification within the same send command are placed in the specified pathname unless the next specification within the send command is another remote station specification (paragraph 3.3.6). If no file or text string is transmitted with the same Send Data (S) command that transmitted the remote receive pathname designation, then all data transmitted by the next send command is placed in the specified remote receive pathname. This specification does not affect subsequent transmissions, unless multiple receive pathnames are specified consecutively, without intervening files or text strings. In this case, the receive pathname specifications are inserted at the beginning of the remote station's receive queue. When multiple pathnames are queued on the remote station's receive queue, each send command dequeues the next receive pathname on the queue and continues dequeueing one pathname for each send command received, until the receiving station's queue is empty. The default receive pathname cannot be dequeued.

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

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**S**	COMMAND:	**S*<$$$$.DIR.FILE>***
PATHNAME:	**<$$$$.DIR.FILE>**		
PATHNAME:	return		

4.5.5 Command File Commands

Command file commands specify a command file as the secondary command source, specify a secondary log destination, and aid in the execution of a command file.

4.5.5.1 Clear Flag Command (CLR). Use the CLR command to clear a specified flag that was set by using the Set Flag command (see paragraph 4.5.5.5).

General Format		*Condensed Format*	
COMMAND:	CLR	COMMAND:	CLR*flag number*
NUMBER:	flag number		

The number of flags required is defined during Emulator installation. The default value is 16 flags, and flags are numbered 0 through >F (0 through 15). Always specify a valid flag number when using the CLR command; otherwise, the message *** INVALID INPUT is logged and the NUMBER: prompt reappears.

All flags are initially set to zero (cleared).

EXAMPLE

General Format	*Condensed Format*
COMMAND: **CLR** NUMBER: **1**	COMMAND: **CLR*1***

4.5.5.2 Command File Command (CF). Use the CF command to specify a pathname (device or file) as the secondary command source.

General Format	*Condensed Format*
COMMAND: CF PATHNAME: pathname	COMMAND: CF*pathname*

Command file commands cannot be nested unless the command file issuing the CF command is the primary command source designated during Emulator activation. When a command file that was opened by a CF command issues a CF command, that file closes and the newly named command file begins issuing commands. Control does not return to the previous command file. Figure 4-1 shows two examples of command file operation, including open and return paths for single and multiple secondary command file sources.

A null (blank) pathname response to the CF command closes the active command file and returns session control to the primary command source at the point following the primary command file's CF command.

The Emulator task terminates if a VDT is active as the primary or secondary command source, and either a hardware error occurs or an end-of-file condition is initiated (by pressing the ENTER key). The task terminates in the same manner as if a TERM or ABORT*TASK command were entered.

EXAMPLE

General Format	*Condensed Format*
COMMAND: **CF** PATHNAME: **.DIR.FILE**	COMMAND: **CF*.DIR.FILE***

Appendix C contains a sample communication session using command files.

4.5.5.3 Conditional Command (IF). Use the IF command to conditionally execute a command or group of commands. When the specified condition is true, the Emulator executes all commands until $ELSE or $END is encountered in the command stream. When $ELSE is encountered, all commands following the $ELSE are skipped until $END is encountered. If the specified condition is false, the Emulator skips all commands until $ELSE or $END is encountered. The Emulator checks the condition only when it executes the IF command, so that subsequent changes in the condition do not alter the path of execution.

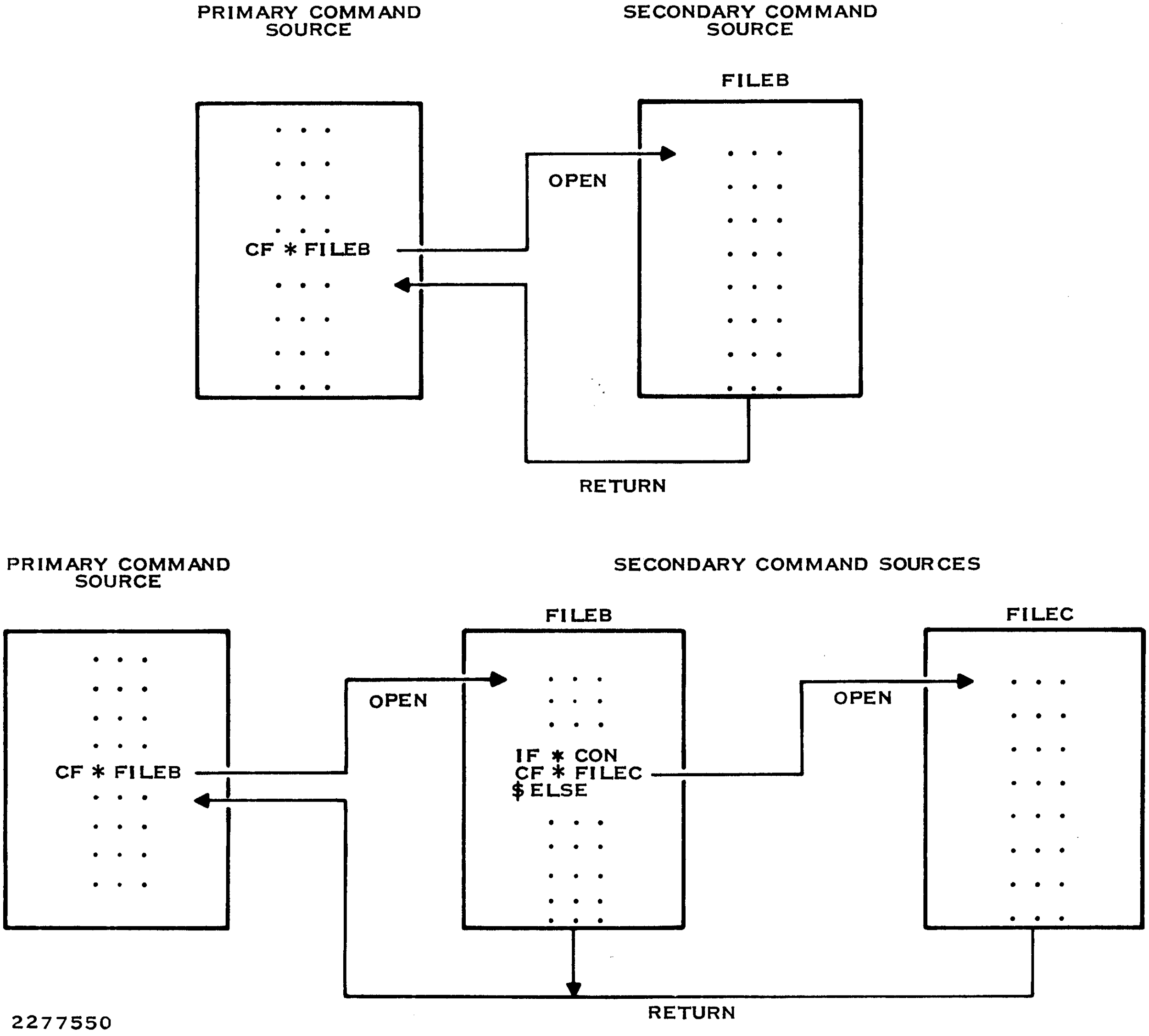


Figure 4-1. Primary and Secondary Command File Operation

The format for all specifiable conditions except RID and LID is as follows:

General Format

COMMAND: IF

OPTION: $\left\{\begin{matrix}\text{condition}\\ ?\end{matrix}\right\}$

Condensed Format

COMMAND: IF* $\left\{\begin{matrix}\text{condition}\\ ?\end{matrix}\right\}$ *

The format for the RID and LID conditions is as follows:

General Format

COMMAND: IF

OPTION: $\left\{\begin{matrix}\text{condition}\\ ?\end{matrix}\right\}$

SEQUENCE: character sequence

Condensed Format

COMMAND: IF* $\left\{\begin{matrix}\text{condition}\\ ?\end{matrix}\right\}$ *character sequence*

Table 4-3 lists the acceptable responses to the OPTION: prompt. 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 and an appropriate message is logged.

The $ELSE and $END are entered as commands in response to the COMMAND: prompt after an IF sequence has been entered. If these commands are entered without beginning an IF sequence, the message *** INVALID INPUT is displayed.

EXAMPLES

General Format	*Condensed Format*
COMMAND: **IF** OPTION: **CON**	COMMAND: **IF*CON***
COMMAND: **IF** OPTION: **DISC**	COMMAND: **IF*DISC***
COMMAND: **IF** OPTION: **LID** SEQUENCE: **ZYLON$5**	COMMAND: **IF*LID*ZYLON$5***

RID also prompts for the sequence.

The following example shows how the commands appear in a command file. The condensed format (see paragraph 4.3.2.2) is recommended for command files.

Condensed Format

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

Table 4-3. Condition Responses to the OPTION: Prompt of the IF Command

Response	Explanation
	The specified condition is true —
CON	If the communication link is connected.
DISC	If the communication link is not connected.
LID	If the local terminal ID matches the identifier given in response to the SEQUENCE: prompt.
RID	If the remote terminal ID matches the identifier given in response to the SEQUENCE: prompt.
S	If the send queue is empty.
RPR	If the receive print queue is empty.[1]
RPU	If the receive punch queue is empty.[1]
ALLQ	If all queues are empty.[1]
EM3780	If this Emulator is a DNOS 3780 Emulator.
EM2780	If this Emulator is a DNOS 2780 Emulator.
SS	If the most recent transmission completes with a completion code of 00 (see Appendix D).
SRPR	If the most recent printer reception completes with a completion code of 00.[2]
SRPU	If the most recent card punch reception completes with a completion code of 00.[2]
SREC	If the most recent send or receive transaction completes with a completion code of 00.
0,1,. . .511	If the specified flag is set.

Notes:

[1] Since the default entry is never removed from the queue and $-option entries are removed only under certain conditions, receive queues are considered empty when the receptions for all entries of normal pathname form are completed.

[2] SRPU and SRPR do not include checking the reception to the default pathname.

4.5.5.4 Log File Command (LF). Use the LF command to specify that all log message output be directed to a new log destination. The response to this command can be any access name acceptable to the operating system. A relative record file or IPC channel must already exist if it is specified in the LF command. The LF command creates a sequential file if a pathname is specified for a file that does not already exist.

General Format	*Condensed Format*
COMMAND: LF	COMMAND: LF*pathname*
PATHNAME: pathname	

Entering the LF command closes the active log file, opens the new log file, and writes a header message as the first record of the file (see paragraph 4.2).

If the LF command is entered with a null response (carriage 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 an error message is written to the journal or, if none is defined, to the system log. The log output reverts to the primary log destination.

EXAMPLE

General Format	*Condensed Format*
COMMAND: **LF**	COMMAND: **LF*.DIR.FILE***
PATHNAME: **.DIR.FILE**	

4.5.5.5 Set Flag Command (SET). Use the SET command to set a specified flag to a value of 1.

General Format	*Condensed Format*
COMMAND: SET	COMMAND: SET*flag number*
NUMBER: flag number	

The number of flags required is defined during Emulator installation. The default value is 16 flags, and flags are numbered 0 through >F (0 through 15). Always specify a valid flag number when using the SET command; otherwise, the message *** INVALID INPUT is logged and the NUMBER: prompt reappears. Initially, the flags are set to zero.

EXAMPLE

General Format	*Condensed Format*
COMMAND: **SET**	COMMAND: **SET*1***
NUMBER: **1**	

4.5.5.6 Wait Command (WAIT). 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. The WAIT command allows delaying operations until the communication line is connected or disconnected or until a specific Emulator queue or all queues are empty.

```
        General Format                        Condensed Format

COMMAND: (WAIT     )              COMMAND:  WAIT*(condition)*
OPTION:  {condition}                             (?        )
         (?        )
```

Table 4-4 lists the acceptable responses to the OPTION: prompt.

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

EXAMPLES

```
        General Format                 Condensed Format

COMMAND:   WAIT                   COMMAND:   WAIT*CON*
OPTION:    CON

COMMAND:   WAIT                   COMMAND:   WAIT*RPR*
OPTION:    RPR
```

4.5.5.7 $ELSE and $END Commands. Use the $ELSE and $END commands in a command file IF-$ELSE-$END structure to achieve conditional skipping of other commands. When the condition specified in the IF statement is true, all succeeding commands are executed until a $ELSE command is encountered. Thereafter, all succeeding commands are skipped until a $END command is encountered. When the condition specified in the IF statement is false, the opposite sequence occurs.

Appendix C includes examples of the use of these commands.

Table 4-4. Condition Responses to the OPTION: Prompt of the WAIT Command

Response	Explanation
RPR	Wait for the receive print queue to empty. Since the default entry may never be removed from the queue and $-option entries are removed only under certain conditions, receive queues are considered empty when all entries of normal pathname form are completed. (Refer to paragraph 2.7 for more information on data receptions.)
RPU	Wait for the receive punch queue to empty (see the RPR option for the definition of empty).
ALLQ	Wait for all queues to empty (see the RPR option for the definition of empty).
S	Wait for the send queue to empty.
CON	Wait until connect is complete. Issue the WAIT command with the CON option only after issuing a command that accomplishes connection; otherwise, the command has no effect.
DISC	Wait for communication link disconnection.

4.5.6 Disconnect Commands

Disconnect commands are available to perform the following functions:

- Abort communications
- Release the telephone line
- Terminate tasks

4.5.6.1 Abort Command (ABORT). Use the ABORT command to initiate one of the following:

- An immediate disconnection
- An immediate disconnection and termination of the Emulator task
- A conditional dequeueing of a specific transaction

Specify the condition by selecting the desired option. The format of the command is as follows and Table 4-5 lists the available options.

General Format

COMMAND: ABORT
OPTION: $\begin{Bmatrix}\text{option}\\ ?\end{Bmatrix}$

Condensed Format

COMMAND: ABORT*$\begin{Bmatrix}\text{option}\\ ?\end{Bmatrix}$*

ABORT Transaction ID (ABORT*nnnnn) does not abort a transaction that is in progress but does remove the transaction from the queue if the transaction is not in progress. To abort a transaction that is in progress, first issue the ABORT*COM command (the transaction is then no longer in progress) and then issue the ABORT Transaction ID command.

Use the ABORT*COM and ABORT*TASK commands to disconnect a remote station if the DISCR command is enabled. When an ABORT*COM or an ABORT*TASK command is executed with DISCR*ON, a DLE EOT is sent to the remote station. When an ABORT*COM or an ABORT*TASK is executed with DISCR*OFF, the Emulator sends only an EOT to the remote station.

CAUTION

Issuing an ABORT*COM with the DISCR command enabled during a file transmission or reception might not disconnect the remote station.

Table 4-5. Responses to the OPTION: Prompt of the ABORT Command

Response	Command Action
COM	Disconnects the communication link and logs a disconnect message. ABORT*COM does not wait for transaction queues to become empty or for any transmission or reception in progress to complete. No transactions are removed from the task queues, and the Emulator task remains active and accepts further command input.
TASK	Terminates the Emulator task, disconnects the communication link, and logs a disconnect message. ABORT*TASK does not wait for transaction queues to become empty or for any transmission or reception in progress to complete. The transaction queues are deleted.
ALLQ	Clears all queues of all transactions. When ABORT*ALLQ is executed while transactions are in progress, the transactions are completed; that is, the completion messages are logged before the queues are cleared.
RPR	Clears the receive print queue of all transactions (except the default). When ABORT*RPR is executed while a receive print transaction is in progress, the reception is completed; that is, the receive print complete message is logged before the queue is cleared.

Table 4-5. Responses to the OPTION: Prompt of the ABORT Command (Continued)

Response	Command Action
RPU	Clears the receive punch queue of all transactions (except the default). When ABORT*RPU is executed while a receive punch transaction is in progress, the reception is completed; that is, the receive punch complete message is logged before the queue is cleared.
S	Clears the send queue of all transactions. When an ABORT*S is executed while a send is in progress, the send is completed and the send complete message is logged before the queue is cleared.
nnnnn	Removes the transaction identifier (nnnnn) specified from the receive or send 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*nnnnn is not performed and a message is logged. Only the significant digits of the transaction identifier need to be entered.

The following examples illustrate the responses required to disconnect (ABORT*COM) and delete a transaction from one of the queues (ABORT*10). In the second example the full queue entry could be (for example) 00010 LP01; that is, only the significant digits of the transaction are required in the OPTION: response when aborting a transaction.

EXAMPLES

General Format		*Condensed Format*	
COMMAND:	**ABORT**	COMMAND:	**ABORT*COM***
OPTION:	**COM**		
COMMAND:	**ABORT**	COMMAND:	**ABORT*10***
OPTION:	**10**		

4.5.6.2 Disconnect Command (DISC). Use the DISC command to disconnect a communications link by performing the equivalent of hanging up the telephone. The following sequence occurs when the DISC command executes:

1. The Emulator waits for all enqueued transmit transactions to be completed.
2. The Emulator waits for any receive transactions in progress to be completed.
3. The Emulator disconnects the communications link.

The general and condensed formats for the command are the same.

Use the DISC command to transmit a disconnect instruction to a remote station if the DISCR command is enabled. When DISCR is enabled, the local station transmits a DLE EOT to the remote station each time the DISC command is executed, indicating that the remote station should disconnect. When the DISCR option is disabled, only an EOT is sent to the remote station each time the DISC command is executed.

EXAMPLE

```
COMMAND: DISC
```

4.5.6.3 Terminate Task Command (TERM). Use the TERM command to terminate the Emulator task. The following sequence occurs when the TERM command executes:

1. The Emulator waits for all enqueued transmit transactions to be completed.
2. The Emulator waits for any receive transactions in progress to be completed.
3. The Emulator now disconnects the communication link.
4. The Emulator terminates the Emulator task.

When DISCR is enabled, executing the TERM command also disconnects the remote station.

The general and condensed formats for this command are the same, as shown in the following example:

EXAMPLE

```
COMMAND: TERM
```

Using the ENTER key has the same effect as using the TERM command.

4.5.7 Miscellaneous Commands

The Reset command, the Statistics command, and the Terminal Identification command perform special functions within the Emulator.

4.5.7.1 Reset Command (RESET). Use the RESET command to cause an end-of-transmission (EOT) signal to be transmitted by the receiving station, via the communication link at the end of the next received block. The EOT resets the communication link to the idle line condition. The transmission and reception of the currently active file is terminated, and the next file, if any, on the send and receive queues is accessed. The general and condensed formats are the same.

EXAMPLE

```
COMMAND: RESET
```

4.5.7.2 Statistics Command (ST). Use the ST command to log (display, if the controlling VDT is also the log) the entire set or a selected subset of the communication statistics. The format of the command is specified as follows and Table 4-6 shows the options available:

```
        General Format                      Condensed Format

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

A null response to the OPTION: prompt or a response of ST* to the COMMAND: prompt specifies the ALL option. Examples of four of the options are given in the following paragraphs. The printouts shown are typical of the statistics that can be displayed and are accompanied by an explanation of some of the entries.

Table 4-6. Responses to the OPTION: Prompt of the ST Command

Response	Command Action
S	Displays/logs a list of the transactions on the send queue.
RPR	Displays/logs a list of the transactions on the receive print queue.
RPU	Displays/logs a list of the transactions on the receive punch queue.
ALLQ	Displays/logs a list of the transactions on all of the queues.
CT	Displays/logs the communication statistics collected since the last communication link connection.
CUR	Displays/logs the current conditions.
ID	Displays/logs the terminal identification information.
ALL	Displays/logs a comprehensive set of statistics (all of the above).
null	Specifies the ALL option (see ALL above).

CUR Option. The following is an example of the CUR option:

EXAMPLE

```
        General Format                Condensed Format

COMMAND:  ST                     COMMAND:  ST*CUR*
OPTION:   CUR
```

Figure 4-2 shows the current condition displays resulting from the use of the ST*CUR command for the 3780 and the 2780 Emulators. The information displayed represents the status of the communication link. It indicates the following:

- Those values set by an Emulator command
- Those values set during installation that have not been varied by a command during this session

In addition to the conditions shown in Figure 4-2, the following information may also be displayed in response to the CUR option:

- Transmit pathname, if transmit is active or pending
- Receive print pathname, if active
- Receive punch pathname, if active

If a transmission or reception is in progress when the statistics are displayed, the status of the communication line is represented as follows:

$$\left\{\begin{array}{c}\text{TRANSMITTING}\\ \text{RECEIVING}\end{array}\right\} \text{ nnnnn xxxxx pathname}$$

where:

nnnnn — is the transaction identification number for transmissions or the receive pathname identification number for receptions.

xxxxx — is the number of records transmitted/received.

pathname — is the pathname of the file being transferred.

CT Option. The following is an example of the format of the CT option:

EXAMPLE

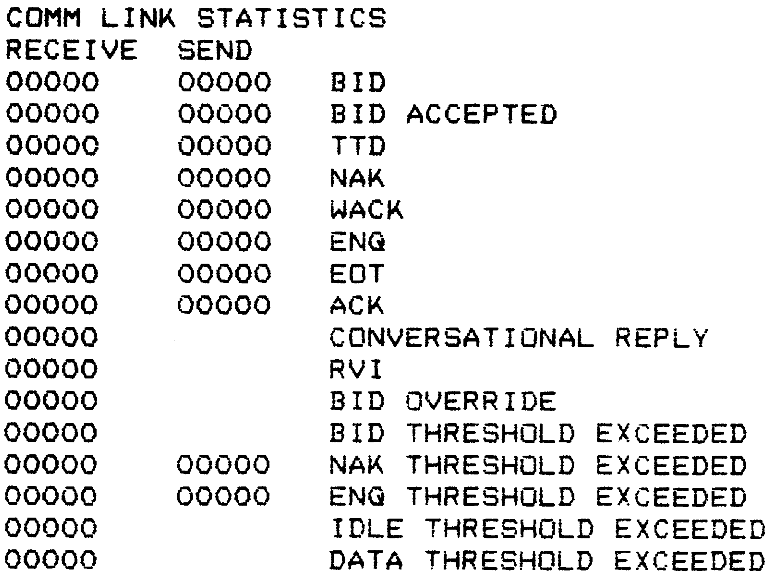
General Format		*Condensed Format*	
COMMAND:	**ST**	COMMAND:	**ST*CT***
OPTION:	**CT**		

```
243 12:22 DNOS/3780  1.0.0 81.001

COMM DISCONNECTED
TRANSPARENCY OFF
COMPRESSION/EXPANSION OFF
RECEIVE PATHNAME OFF
SEND REQUEST OFF
PRINT FILE VFC OFF
BID TASK OFF
OPERATOR MESSAGE OFF
RETAIN TRAILING BLANKS OFF
DISCONNECT REMOTE OFF
BLOCK LENGTH 00512
SEND RECORD LENGTH 00080
RECEIVE PRINT RECORD LENGTH 00132
RECEIVE PUNCH RECORD LENGTH 00080
PRIMARY COMMAND ACCESS NAME  ST04
SECONDARY COMMAND ACCESS NAME   .DIR.FILE
PRIMARY LOG ACCESS NAME  ST04
SECONDARY LOG ACCESS NAME  LP01
JOURNAL ACCESS NAME  .JOURNAL
COMM DEVICE NAME  CM02
FLAGS 00000 TO 00015:  0000 1110 0000 0000


248 15:59 DNOS/2780  1.0.0 81.001

COMM DISCONNECTED
TRANSPARENCY OFF
END OF MEDIA OFF
RECEIVE PATHNAME OFF
SEND REQUEST OFF
PRINT FILE VFC OFF
BID TASK OFF
OPERATOR MESSAGE OFF
RETAIN TRAILING BLANKS OFF
DISCONNECT REMOTE OFF
BLOCK LENGTH 00400
SEND RECORD LENGTH 00080
RECEIVE PRINT RECORD LENGTH 00132
RECEIVE PUNCH RECORD LENGTH 00080
RECORDS PER BLOCK 00007
PRIMARY COMMAND ACCESS NAME  ST04
SECONDARY COMMAND ACCESS NAME  .DIR.FILE
PRIMARY LOG ACCESS NAME  ST04
SECONDARY LOG ACCESS NAME  LP01
JOURNAL ACCESS NAME  .JOURNAL
COMM DEVICE NAME  CM02
FLAGS 00000 TO 00015:  0000 1110 0000 0000
```

Figure 4-2. Current Conditions Displays for the DNOS 3780/2780 Emulators

The information displayed in response to the command ST*CT shows statistics accumulated since the last communication link was established. The statistics display shown in Figure 4-3 pertains to the DNOS 3780 Emulator. The corresponding display for the DNOS 2780 Emulator does not include entries for TTD and WACK since these are not part of 2780 operation. The statistics shown refer to binary synchronous communications (BSC) procedures (see Appendix E) and reflect the information given in Table 4-7. The counts are reset to zero with each new request for a connection.

If the ST*CT, ST*, or ST*ALL command is executed following a disconnection, the statistics display continues to show the values present at the termination of the connection. Statistics counters remain constant until the next connection, but are not reset. They increment following transmissions/receptions. These counters are only reset between task activations.



```
COMM LINK STATISTICS
RECEIVE  SEND
00000    00000    BID
00000    00000    BID ACCEPTED
00000    00000    TTD
00000    00000    NAK
00000    00000    WACK
00000    00000    ENQ
00000    00000    EOT
00000    00000    ACK
00000             CONVERSATIONAL REPLY
00000             RVI
00000             BID OVERRIDE
00000             BID THRESHOLD EXCEEDED
00000    00000    NAK THRESHOLD EXCEEDED
00000    00000    ENQ THRESHOLD EXCEEDED
00000             IDLE THRESHOLD EXCEEDED
00000             DATA THRESHOLD EXCEEDED
```

Figure 4-3. DNOS Emulator Communication Link Statistics Display

Table 4-7. Communication Link Statistics

Parameter	Explanation
BID	Specifies the number of BIDs; this is a count of the ENQs received and sent as bids for the use of the communication line.
BID ACCEPTED	Specifies the number of BIDs accepted; this is a count of the positive responses received and transmitted in response to a bid sequence. Local Remote Receive BID → ← ACK0 Transmit ← BID ACK0 →
TTD	Specifies the number of TTDs; this is a count of the TTD sequences received and transmitted. The transmitting station sends the TTD sequence to indicate that it is temporarily not ready to transmit. The 2780 Emulator cannot transmit a TTD. Local Remote Receive ← Data ACK0 → ← TTD NAK → Transmit Data → ← ACK0 TTD → ← NAK
NAK	Specifies the number of NAKs; this is 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.

Table 4-7. Communication Link Statistics (Continued)

Parameter	Explanation
WACK	Specifies the number of WACKs; this is a count of the number of WACKs received and transmitted. The receiving station sends the WACK to indicate that it is temporarily not ready to receive. The 2780 Emulator cannot transmit a WACK.

Local		Remote
	Receive	
Data	→	
	←	WACK
	Transmit	
	←	Data
WACK	→	

Parameter	Explanation
ENQ	Specifies the number of ENQs; this is a count of the ENQs received and transmitted. The station sends the ENQ to bid for the line or as part of the TTD sequence.
EOT	Specifies the number of EOTs; this is a count of the EOTs received and transmitted. The EOT is transmitted to terminate a transmission or to indicate error conditions.
ACK	Specifies the number of ACKs; this is a count of the ACKs received and transmitted. An ACK is transmitted as a positive response to a text block (that is, the block was received without error) or in response to a line bid.
CONVERSATIONAL REPLY	Specifies the number of conversational replies received. The host can respond with a conversational reply in place of a positive acknowledgement after it receives an end-of-text (ETX) (3780 Emulator users only). The conversational reply data is directed to the print queue.

Local		Host
Data	→	
	←	ACK0/ACK1
Data/ETX	→	
	←	Conversational Reply (data)
ACK1/ACK0	→	

Table 4-7. Communication Link Statistics (Continued)

Parameter	Explanation
RVI	Specifies the number of 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. The capability to transmit an RVI is optional on host systems. The Emulator cannot transmit an RVI but will process received RVIs. Local — Host Data → ← RVI Data → ← ACK0 EOT → ← BID ACK0 → ← Data . . ACK1 → . . . . . ← Data/ETX ACK0 → ← EOT BID → ← ACK0 Data →
BID OVERRIDE	Specifies the number of 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. The bid response time-out value (specified at installation) for a station designated as primary should be different from that of a station designated as secondary to avoid a call collision deadlock. Local — Remote BID →← BID ← BID ACK0 →
BID THRESHOLD EXCEEDED	Specifies the number of times the BID threshold has been exceeded.* The line is bid for n times in a row without receiving a bid accepted response or data. The value of n is the threshold.
NAK THRESHOLD EXCEEDED	Specifies the number of times the NAK threshold has been exceeded.* The station responds with/transmits NAK n times in a row. The value of n is the threshold.

Table 4-7. Communication Link Statistics (Continued)

Parameter	Explanation
ENQ THRESHOLD EXCEEDED	Specifies the number of times the ENQ threshold has been exceeded.* The station has responded with/transmitted ENQ n times in a row. The value of n is the threshold.
IDLE THRESHOLD EXCEEDED	Specifies the number of times the IDLE threshold has been exceeded.* The station has not received any data or line activity from the remote station for n seconds. The value of n is the threshold.
DATA THRESHOLD EXCEEDED	Specifies the number of times the DATA threshold has been exceeded.* The station has not received data for n time-out intervals. These intervals include intercharacter time-outs and time-outs between blocks. The value of n is the threshold.

Note:

* These thresholds are set during installation and building of the Emulator, as described in Appendix B.

ALLQ Option. The following is an example of the format of the ALLQ option:

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**ST**	COMMAND:	**ST*ALLQ***
OPTION:	**ALLQ**		

The information displayed in response to the ST*ALLQ command gives statistics on the status of the Emulator queues. An example follows:

```
SEND QUEUE
EMPTY

RECEIVE PRINT QUEUE
00001 LP01

RECEIVE PUNCH QUEUE
00002 DUMY
```

The format of the ST*ALLQ display is as follows:

```
{xxxxx   QUEUE                }
{nnnnn   pathname. . .pathname}
```

where:

xxxxx — is SEND, RECEIVE PRINT, or RECEIVE PUNCH.

nnnnn — is the identification number of the transaction or of the receive pathname.

pathname — is the pathname(s) 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 S queue is specified and it has no entries, the message EMPTY appears.

ID Option. The following is an example of the format of the ID option:

EXAMPLE

General Format		*Condensed Format*	
COMMAND:	**ST**	COMMAND:	**ST*ID***
OPTION:	**ID**		

The information displayed in response to the ST*ID command gives both remote and local terminal ID information, as shown in the following example:

```
TERMINAL IDENTIFICATION
REMOTE TERMINAL ID: CENTURION
LOCAL TERMINAL ID:  ZYLON$5
```

The REMOTE TERMINAL ID: prompt identifies the source of received data block (that is, the remote station). LOCAL TERMINAL ID: is the identifier that precedes the first block in a transmission from the local station. The ID command lists the currently assigned values for these identifiers. If terminal identification is not enabled, no values are assigned to the identifiers. Terminal identification is enabled during installation or by the use of the Terminal Identification (ID) command.

4.5.7.3 Terminal Identification Command (ID). Use the ID command to specify an identifier that is to precede the first transmission from the local station. The last ID assigned prior to a connection becomes the ID for that connection. If the ID is altered during the session, the alteration does not take effect until the next connection, but it is displayed as the local terminal ID.

Generally, the ID command is used only with switched telephone lines.

EXAMPLE

General Format	*Condensed Format*
COMMAND: ID	COMMAND: ID*identifier*
SEQUENCE: identifier	

The identifier may be 1 to 20 characters long. The characters can include EBCDIC characters in the range >41 through >FF (ASCII >21 through >7F) except for blanks, asterisks, and ampersands.

When communicating with an IBM 3780, the Emulator expects a maximum of five characters as an identifier. The first two characters of the identifier must be identical and must be ASCII characters translatable to EBCDIC within the range >C0 through >FF (uppercase alphabetic, numeric, and special characters). The rest of the characters may be any combination of noncontrol characters (except blank, asterisk, and ampersand) in the range >41 through >FF.

When communicating with an IBM 2780, the Emulator expects a two-character identifier. The character that is repeated in the sequence is in EBCDIC within the range >80 through >FF (uppercase and lowercase alphabetic, numeric, and special characters).

Responding to SEQUENCE: with a null response clears the identifier previously set and disables sending a terminal identification. No value is logged for the local terminal ID with a subsequent ST*ID.

A terminal identification may have been specified during Emulator installation. To change this ID, use the ID command. If no identification was specified during installation but the response to the ENABLE TERMINAL IDENTIFICATION?: prompt was YES, a default identification of 20 zeros is used as the terminal identification. If the response to the ENABLE TERMINAL IDENTIFICATION?: prompt was NO, no identification is transmitted. The ID command overrides a response of NO to the prompt.

EXAMPLE

General Format	*Condensed Format*
COMMAND: **ID**	COMMAND: **ID*DN2780***
SEQUENCE: **DN2780**	

4.5.7.4 Bid Command (BID). Use the BID command to specify a local task for execution. (Compare BID to the remote bid task feature.)

EXAMPLE

General Format

COMMAND: BID

PROGRAM: progname

STATION: $\left\{\begin{matrix}\text{return}\\ \text{terminal id}\\ \text{synonym}\end{matrix}\right\}$

EXECUTE: $\left\{\begin{matrix}\text{YES}\\ \text{NO}\\ \text{?}\end{matrix}\right\}$

PARAMETERS: parm

Condensed Format

COMMAND: BID*progname*$\left\{\begin{matrix}\text{return}\\ \text{terminal id}\\ \text{synonym}\end{matrix}\right\}$*$\left\{\begin{matrix}\text{YES}\\ \text{NO}\\ \text{?}\end{matrix}\right\}$*parm*

The response to the PROGRAM: prompt is the name of the program to be executed and must be a valid program name installed in the .S$SHARED program file. The program name can be from one to eight characters long; the first character must be alphabetic, and the rest can be alphabetic or numeric.

If the task is required to execute at an interactive terminal, respond to the STATION: prompt with the identification (device type and number) or synonym of the terminal. If the task is required to execute independently of a terminal, default through the STATION: prompt by pressing the RETURN key. Specifying a terminal identification causes that terminal to be in use by the task until the task completes or is aborted. If the terminal selected is already in use, an error is returned when the bid is attempted.

A YES response to the EXECUTE: prompt causes the specified task to execute concurrently with the Emulator; a NO response suspends the Emulator task until the specified task terminates.

The response to the PARAMETERS: prompt can be up to four ASCII characters that specify parameters to be passed to the bid task. The bid task must issue a Get Parameters SVC to obtain the parameters. If there are no parameters, default through the PARAMETERS: prompt.

When the specified task is bid, the following message appears:

```
(date/time) 5005 TASK BID PROCESSED xx
```

where:

xx is a completion/error code.

A code of 00 indicates that the task is successfully bid.

A code of 59 indicates that incorrect syntax was encountered during prompt response processing. Other error codes are described in the *Model 990 Computer DNOS Messages and Codes Reference Manual*.

EXAMPLES

```
        General Format                      Condensed Format

COMMAND:    BID                 COMMAND:  BID*EXECDST*ST04*YES*2AM*
PROGRAM:    EXECDST
STATION:    ST04
EXECUTE:    YES
PARAMETERS: 2AM

COMMAND:    BID                 COMMAND:  BID*MYTSK**NO**
PROGRAM:    MYTSK**NO**
```

Appendix A

Emulator Hardware Requirements

The Emulator runs with the DNOS standard equipment configuration. The following minimum equipment is required:

- A Model 990/10 or 990/12 computer equipped with:
  - 128K words (256K bytes) of memory
  - A system disk
  - A system console which can be either a VDT or an ASR data terminal
  - A means of disk backup
- One of the communications interface sets listed in Table A-1 (for TI modems) or Table A-2 (for non-TI modems)

The supported non-TI modems are the Bell (Western Electric) modems listed in Table A-3. The options recommended are listed with each modem. Table A-3 also lists those modem options that are required when the specified modem is used with the Emulator. The non-TI communications components (Table A-2) support the Bell modems (Table A-3).

Use cassette tape for Emulator I/O only if the 733 ASR is not also the Emulator operator console. I/O using slow devices (that is, 733 ASR or cassette tape) may cause error messages on the host console. Refer to the *Model 990 Computer DNOS Operations Guide* for information on DNOS supported hardware.

Many telephone companies now offer facilities capable of transmitting digital information directly without converting it to analog. A modem is not required for digital service. However, a similar device called a data service unit (DSU) provides the interface between the communications controller and the communications line. Digital data service (DDS) is generally used only for high-speed information exchange between computers and is not yet available in all areas.

A major advantage of digital service is transmission without the signal degradation experienced in analog transmission. Digital transmission occurs without amplification of line noise and distortion passed by in-line signal amplifiers as is the case in analog transmission.

DSUs are available to replace existing analog modems using RS-232C interface at 2400, 4800, and 9600 bps for 3780/2780 transmissions. Consult your Bell sales and service representative for information regarding the use of DSUs in your area.

Table A-1. TI Communications Interface Kits

Communications Interface	TI Modem (201C equivalent)	Auto-Call Unit (ACU) (optional)
Communications Interface Module (CIM) Kit: 946104-0001	Synchronous Modem Kit: 945094-0003	ACU Kit: 945163-0001
Four-Channel Communications Controller (FCCC) Kit: 2261954-0001*	Synchronous Modem Kit: 945094-0003 (one per channel)	ACU Kit: 945163-0001
Bit-Oriented, Character-Oriented, Asynchronous Interface Module (BCAIM) Kit: 2265184-0001	Synchronous Modem Kit: 945094-0004	ACU Kit: 945163-0001

Note:

* The revision level for the FCCC controller in this kit should be Rev. P or later.

Table A-2. Non-TI Communications Interface Components

Communications Interface	Non-TI Modem Components	External Auto-Call Unit Interface[1]
Communications Interface Module (CIM) Kit: 946104-0002	The kit comprises: CIM — 946105-0001 Cable — 946117-0001	The following kit is required for all 3 interfaces:
Four-Channel Communications Controller (FCCC): 2261955-0002[2]	The FCCC requires: Cable — 946117-0001 or 0002 (one per channel)	EACUI Kit: 2263907-0001
Bit-Oriented, Character-Oriented, Asynchronous Interface Module (BCAIM) Kit: 2265184-0002	The kit comprises: BCAIM — 2265165-0001 Cable — 2303070-0002 or 2303079-0002	The kit comprises: EACUI — 2263480-0001 Cable — 2263477-0001

Note:

[1] Tandem dialing is not supported for auto-call units used with non-TI modems.

[2] The revision level for the FCCC should be Rev. P or later.

Table A-3. Supported Non-TI Modems

Western Electric (Bell) Modem	Recommended Options	Required Options
Model 201A switched network operation, 2000 bps	Half-duplex Alternate voice Selective unattended answer Without New Sync RTS carrier control	EIA option Internal timing
Model 201B leased-line operation, 2400 bps	None	EIA option Internal timing Without New Sync
Model 201C switched network operation, 2400 bps	Permanent auto-answer Grounding option (AB not connected to AA)	Internal timing EIA voltages for CD, RDY, and ring
Model 201C leased-line operation, 2400 bps	Grounding option (AB not connected to AA)	Without New Sync Internal timing
Model 208A leased-line operation 4800 bps	Grounding option (AB not connected to AA)	Internal timing
Model 208B switched network operation, 4800 bps	Grounding option (AB not connected to AA) Auto-answer by CD control	Internal timing
Model 209A multiplexed leased-line operation, 2400, 4800, 7200, 9600 bps	Grounding option (AB not connected to AA) Elastic store OUT	Internal timing
Model 212A switched network operation, 300, 1200 bps	Data set ready indication (CC) Auto-answer IN Signal/frame grounds connection OUT Common CB (CTS) and CF (DCD) indications Carrier loss disconnect OUT Make busy CN circuit OUT	Internal timing 1200 bps

Table 4-3. Supported Non-TI Modems (Continued)

Western Electric (Bell) Modem	Recommended Options	Required Options
Model 2024A[1] DATAPHONE II leased-line operation, 2400 bps	None	Internal timing
Model 2048A[1] DATAPHONE II leased-line operation, 4800 bps	None	Internal timing
Model 2096A[1] DATAPHONE II leased-line operation, 2400, 4800, 9600 bps	None	Internal timing

Note:

[1] DATAPHONE II modems differ from other models because they have a diagnostic console not found on other Bell modems.

Appendix B

Emulator Installation

B.1 INTRODUCTION

The installation of the DNOS 3780 and 2780 Emulators requires the following:

- Generating the DNOS system
- Building and installing the appropriate device service routine (DSR) and task

The Emulator package includes batch streams and necessary link control files to aid these installation processes.

B.2 GENERATING THE DNOS SYSTEM

For information on generating the DNOS operating system, refer to the *Model 990 Computer DNOS System Generation Manual* and the *Model 990 Computer DNOS 3780/2780 Emulator Object Installation*.

B.3 BUILDING AND INSTALLING THE DNOS DSR AND TASK

In building the DSR and task, either use the standard configuration and protocol parameters or modify them by using the BCD and BET commands.

For further information about building and installing the Emulator DSR and task, refer to the *Model 990 Computer DNOS 3780/2780 Emulator Object Installation*.

Tables B-1 and B-2 list the Emulator parameters that can be modified during installation by using the BCD and BET commands. The tables include standard or default values for these parameters and provide space for recording user-defined values.

The DSR parameters listed in Table B-1 are set during installation and cannot be modified after activation of the Emulator. If changes are needed in the DSR configuration, it is necessary to reinstall the Emulator. Some of the task parameters listed in Table B-2 are modifiable after building the DSR (that is, after activating the Emulator) and are marked as such in the table. (See paragraph 4.5 for further details.)

Table B-1. DNOS DSR Installation Modifiable Parameters

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]
Board type[2]	3780 2780	COMIF, FCCC, or BCAIM	COMIF	__________
ACU CRU address[2]	3780 2780	>FFFF = no ACU available >xxxx = correct ACU CRU address	>FFFF	__________
Dial type[2]	3780 2780	0 = touch-tone 1 = rotary dial (pulse)	1	__________
Bid Response time-out[2]	3780 2780	Zero (infinite) through 60 seconds	01	__________
Receive ACK time-out[2]	3780 2780	Zero (infinite) through 60 seconds	03	__________
Receive Data time-out[2] (for host station operation, set this value to about 30 seconds)	3780 2780	Zero (infinite) through 60 seconds	00	__________
Idle line timeout[2]	3780 2780	Zero (infinite) through 65,535 1-second intervals	20	__________
Transmit BID threshold[2]	3780 2780	Zero (infinite) through 65,535 counts	15	__________
Receive ENQ threshold[2]	3780 2780	Zero (infinite) through 65,535 counts	15	__________
Transmit ENQ threshold[2]	3780 2780	Zero (infinite) through 65,535 counts	15	__________
Receive NAK threshold[2] (for host station operation, set this value to 15)	3780 2780	Zero (infinite) through 65,535 counts	00	__________

Table B-1. DNOS DSR Installation Modifiable Parameters (Continued)

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]
Transmit NAK threshold[2] (for host station operation, set this value to 15)	3780 2780	Zero (infinite) through 65,535 counts	00	
Receive data threshold[2]	3780 2780	Zero (infinite) through 65,535 counts	20	

Notes:

[1] This column is provided to record user-defined values set during the BUILDCOM process.

[2] This parameter cannot be modified after installation.

Table B-2. DNOS Task Installation Modifiable Parameters

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]
Receive printer pathname (RPRD, 4.5.1.8)	3780 2780	Any valid pathname (including $-option)	LP01	
Receive punch pathname (RPUD, 4.5.1.9)	3780 2780	Any valid pathname (including $-option)	DUMY	
Number of user flags[2]	3780 2780	0 through 511 flags	16	
Pending receive and send requests[2]	3780 2780	Any number that, when multiplied by 2.5, yields a whole number[3]	8	

Notes:

[1] This column is provided to record user-defined values set during the BUILDCOM process.

[2] This parameter cannot be modified after installation.

[3] This value also depends on system memory constraints.

Table B-2. DNOS Task Installation Modifiable Parameters (Continued)

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]

Transparency ON/OFF (TY, 4.5.1.15)	3780
	2780
		NO = disabled
		YES = enabled
			NO	__________

Disconnect on error[2]	3780
	2780
		NO = do not disconnect
		YES = disconnect
			NO	__________

Disconnect remote (DISCR, 4.5.1.4)	3780
	2780
		NO = disabled
		YES = enabled
			NO	__________

Call connect time-out[2]	3780 2780	Zero (infinite) through 255 minutes	0	__________
Auto-call retry count[2]	3780 2780	No minimum/maximum	3	__________
Call retry time (seconds)[2]	3780 2780	No minimum/maximum	3	__________
Interdigit time-out (100 ms)[2]	3780 2780	No minimum/maximum	10	__________
Dial tone time-out (100 ms)[2]	3780 2780	No minimum/maximum	10	__________
Present next digit time-out (100 ms)[2]	3780 2780	Zero (no delay) through 65,535 intervals of 100 ms	0	__________

Remote receive pathname ON/OFF (RPN, 4.5.1.6)	3780
	2780
		NO = disabled
		YES = enabled
			NO	__________

Remote receive pathname code[2]	3780 2780	Any legal string of 4 characters in single quotes	'$$$$'	__________

Vertical format control (VFC, 4.5.1.16)	3780
	2780
		NO = disabled
		YES = enabled
			NO	__________

Notes:

[1] This column is provided to record user-defined values set during the BUILDCOM process.

[2] This parameter cannot be modified after installation.

Table B-2. DNOS Task Installation Modifiable Parameters (Continued)

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]

Send request ON/OFF
(SRQ, 4.5.1.14)
	3780
	2780
		NO = disabled
		YES = enabled
			NO	

Send request code[2]	3780 2780	Any legal string of 4 characters in single quotes	'$$$S'	

Trailing blanks
(TRAILB, 4.5.1.12)
	3780
	2780
		NO = disabled
		YES = enabled
			NO	

Remote bid task
(TBID, 4.5.1.1)
	3780
	2780
		NO = disabled
		YES = enabled
			NO	

Remote bid task code[2]	3780 2780	Any legal string of 4 characters in single quotes	'$BID'	
Operator message code[2]	3780 2780	Any legal string of 4 characters in single quotes	'$OPR'	

Operator message
($OPR 4.5.1.17)
	3780
	2780
		NO = disabled
		YES = enabled
			NO	

$-Option character[2]	3780 2780	$ or # enclosed in single quotes	'$'	
Maximum transmit record length[2]	3780 2780	80 through 1021 bytes	256	
Maximum receive record length[2]	3780 2780	80 through 1021 bytes	256	
Send record length (SRL, 4.5.1.13)	3780 2780	80 through specified maximum bytes[4]	80	
Receive punch record length (RPURL, 4.5.1.10)	3780 2780	80 through specified maximum bytes [5]	80	

Notes:

[1] This column is provided to record user-defined values set during the BUILDCOM process.

[2] This parameter cannot be modified after installation.

[4] The maximum value is determined by the value set for the maximum transmit record length.

[5] The maximum value is determined by the value set for the maximum receive record length.

Table B-2. DNOS Task Installation Modifiable Parameters (Continued)

Parameter and Change Default Command (if any)	Emulator	Value Range	Default Value	User-Defined Value[1]
Receive print record length (RPRRL, 4.5.1.7)	3780 2780	80 through specified maximum bytes[5]	132	__________

Enable terminal ID option[2]	3780
	2780
		NO = disabled
		YES = enabled
			NO	__________

Terminal ID (ID, 4.5.7.3)	3780 2780	Up to 20 characters in the range >41 through >FF (EBCDIC) or >21 through >7F (ASCII)	0	__________

Compression/expansion ON/OFF (CE, 4.5.1.3)	3780	NO = disabled
		YES = enabled
			NO	__________

Block length for 3780 (BL, 4.5.1.2)	3780	80 through 1024 bytes	512	__________
Punch device select[2]	3780	DC2 or DC3	DC2	__________

End-of-media (EM, 4.5.1.5)	2780	NO = disabled
		YES = enabled
			NO	__________

Block length for 2780 (BL, 4.5.1.2)	2780	80 through 1024 bytes	400	__________
Records per block (RB, 4.5.1.11)	2780	Minimum = 1 record Maximum = as many records as the block will hold	7	__________

Notes:

[1] This column is provided to record user-defined values set during the BUILDCOM process.

[2] This parameter cannot be modified after installation.

[5] The maximum value is determined by the value set for the maximum receive record length.

Appendix C

Communication Session

C.1 INTRODUCTION

The following is a sample communication session in which four command files (.CMDFIL0 through .CMDFIL3) are used at a local (host) station to call two remote stations, site 1 and site 2. Each remote station uses two command files (.REMOTE1 and .REMOTE2) to receive the transmissions from the host. When a successful connection has been accomplished, a specified file is sent to each remote station.

The main functions of each command file at the host station are as follows:

- .CMDFIL0 — Instructs .CMDFIL1 to call remote site 1.
- .CMDFIL1 — Calls remote site 1 and sends a file.
- .CMDFIL2 — Calls remote site 2 and sends a file.
- .CMDFIL3 — Checks whether .CMDFIL1 and .CMDFIL2 completed successfully; if not, it restarts the process. If successful, it terminates the Emulator task.

The main functions of each command file at each remote station are as follows:

- .REMOTE1 — Answers a call placed by the host to this site and receives the transmitted file.
- .REMOTE2 — Checks to see if the reception was successful; if not, it reinvokes .REMOTE1. If successful, it terminates the Emulator task.

C.2 LOCAL STATION COMMAND FILES

The local station uses four command files to control the communication session.

C.2.1 Host Control Command File

This file sets flags 1 and 2 to indicate that a call must be placed to remote sites 1 and 2, respectively.

```
            COMMAND FILE:  .CMDFIL0                    SITE:  HOST

   SET*1*                   SET FLAG TO CALL REMOTE SITE 1;
   SET*2*                   SET FLAG TO CALL REMOTE SITE 2;
   CF*.CMDFIL1*             EXECUTE .CMDFIL1 TO CALL SITE 1.
```

C.2.2 Command File One

This file calls and sends a file to remote site 1. The file sets flag 1 if either a connection to the remote site could not be made or the transmission of the file was unsuccessful.

When this command file is first accessed from .CMDFIL0, the $END is not paired with an IF. The Emulator expects a command at this point, and outputs '*** INVALID INPUT' to the log file upon reading $END. The Emulator then reads the next record of the command file and executes the IF*1*. This error does not terminate the command stream. The unpaired $END in .CMDFIL1 is required because .CMDFIL3 conditionally specifies .CMDFIL1 as the new secondary command file. When the condition is met, control is transferred, and the Emulator does not read the $END paired with the IF in .CMDFIL3.

```
               COMMAND FILE:  .CMDFIL1                  SITE:   HOST

     $END                      CLOSES IF FROM .CMDFIL3
     IF*1*                     IF FLAG 1 IS SET, CALL SITE 1;
     CLR*1*                    CLEAR THE FLAG;
     CALL*1-512-250-7407       AUTOMATICALLY DIAL SITE 1;
     WAIT*CON*                 WAIT FOR A CONNECTION.
     $END
```

If the connection to site 1 is established, send the file .DIR.FILE and wait for the transmission to complete successfully. If the connection is unsuccessful, set flag 1 to reinvoke this command.

```
     IF*CON*                   IF A CONNECTION TO SITE 1 IS ESTABLISHED
     S*.DIR.FILE*              SEND FILE .DIR.FILE;
     WAIT*S*                   WAIT FOR THE TRANSMISSION TO COMPLETE.
     $ELSE                     CONNECTION FAILED SO
     SET*1*                    SET FLAG 1 TO TRY AGAIN.
     $END
```

If the transmission to site 1 completes without error, disconnect the line and call site 2. If the transmission fails, set flag 1 to reinvoke this command file.

```
     IF*SS                     WAS THE FILE TRANSMISSION SUCCESSFUL?
     $ELSE                     NO
     SET*1*                    SO SET FLAG 1 TO TRY AGAIN.
     $END
     DISC*                     YES, SO DROP THE LINE.
     IF*2*                     IF FLAG 2 IS SET
     CF*.CMDFIL2*              EXECUTE .CMDFIL2 TO CALL SITE 2.
     $ELSE
     CF*.CMDFIL3*              OTHERWISE EXECUTE .CMDFIL3 AND FINISH
     $END
```

C.2.3 Command File Two

This command file calls and sends a file to site 2. The file sets flag 2 if either a connection to the site could not be made or the file transmission was unsuccessful.

```
          COMMAND FILE:  .CMDFIL2                  SITE:  HOST

    $END                 CLOSES IF FROM .CMDFIL1
    IF*2*                IF FLAG 2 IS SET, CALL SITE 2;
    CLR*2*               CLEAR THE FLAG;
    CALL*258-4888*       AUTOMATICALLY DIAL SITE 2.
    WAIT*CON*            WAIT FOR A CONNECTION.
    $END
```

If the connection to site 2 is established, send the file .DIR.FILE and wait for the transmission to complete successfully. If the connection is unsuccessful, set flag 2 to reinvoke this file.

```
    IF*CON*               IF A CONNECTION IS ESTABLISHED
    S*.DIR.FILE*          SEND FILE .DIR.FILE;
    WAIT*S*               WAIT FOR THE TRANSMISSION TO COMPLETE.
    $ELSE*                CONNECTION FAILED SO
    SET*2*                SET FLAG 2 TO TRY AGAIN.
    $END
```

If the transmission to site 2 completes without error, disconnect the line. If the transmission fails, set flag 2 to reinvoke this file.

```
    IF*SS*                  WAS THE TRANSMISSION SUCCESSFUL?
    $ELSE*                  NO
    SET*2*                  SO SET FLAG 2 TO TRY AGAIN.
    $END
    DISC*                   YES, SO DROP THE LINE.
    CF*.CMDFIL3*            EXECUTE .CMDFIL3 TO
                              VERIFY SUCCESSFUL OPERATION
```

C.2.4 Command File Three

This command file verifies that .CMDFIL1 and .CMDFIL2 completed their functions successfully. If site 1 failed to receive the transmitted file, .CMDFIL1 is reinvoked. If site 2 failed to receive the transmitted file, .CMDFIL2 is reinvoked. If both sites received the file, Emulator execution terminates.

```
          COMMAND FILE:  .CMDFIL3                    SITE:  HOST

    $END                 CLOSES IF FROM .CMDFIL2
    IF*1*                IF SITE 1 NEEDS A RETRY
    CF*.CMDFIL1*         CALL .CMDFIL1
    $END
    IF*2*                IF SITE 2 NEEDS A RETRY
    CF*.CMDFIL2*         CALL .CMDFIL2
    $END
    TERM*                IF FLAGS 1 AND 2 ARE OFF
                         TERMINATE THE EMULATOR
```

C.3 REMOTE STATIONS (SITES 1 AND 2) COMMAND FILES

Both remote stations use the command files .REMOTE1 and .REMOTE2.

C.3.1 Remote Command File One

This command file answers a call placed by the host to this remote site. This file waits for a connection to be established and then waits to receive a file transmitted by the host. If the reception fails, the Emulator waits for another call from the host. If the reception is successful, the Emulator terminates.

```
          COMMAND FILE:  .REMOTE1                 SITE:   REMOTE

    RPR*.FILE*               SET UP PATHNAME OF FILE TO RECEIVE DATA;
    CLR*0*                   CLEAR RECEPTION SUCCESSFUL FLAG;
    ANS                      SET UP AUTOMATIC ANSWER CONDITION;
    WAIT*CON*                WAIT FOR THE PHONE TO RING;
```

If the connection is established, receive the file into .FILE. Wait for a successful completion; if the connection is unsuccessful, set flag 0 to reinvoke this file.

```
    IF*CON*                   IF A CONNECTION IS ESTABLISHED
    WAIT*RPR*                 WAIT FOR FILE RECEPTION TO COMPLETE;
    $ELSE                     OTHERWISE
    SET*0*                    SET FLAG 0.
    $END
```

If the reception completes successfully, disconnect the line and terminate the Emulator. If the reception fails, set flag 0 to reinvoke this command.

```
    IF*SRPR*                  WAS FILE RECEPTION SUCCESSFUL?
    $ELSE*                    NO; SO
    SET*0*                    SET FLAG 0 FOR RETRY.
    $END
    DISC*                     YES; SO DROP THE LINE
    CF*.REMOTE2*              EXECUTE .REMOTE2 FOR CHECK.
```

C.3.2 Remote Command File Two

This command file checks flag 0. If it is set, the reception failed and .REMOTE1 must be reinvoked. If the flag is reset, the Emulator terminates.

```
          COMMAND FILE:  .REMOTE2                 SITE:   REMOTE

    IF*0*                    IS THE RETRY FLAG SET?
    CLR*0*                   YES, SO CLEAR IT FOR RETRY.
    $ELSE*                   NO, SO THE FILE WAS SUCCESSFULLY RECEIVED AND
    TERM                     THE EMULATOR CAN BE TERMINATED.
    $END
    CF*.REMOTE1              EXECUTE .REMOTE1 FOR RETRY.
                             REMOTE1 CANNOT BE PRIMARY COMMAND SOURCE
```

Appendix D

Emulator Messages

This appendix lists and explains the messages that the Emulator displays. Several of these messages include a completion/error code. Table D-1 describes completion/error codes that may be generated during Emulator operation.

The messages are listed numerically according to their message number. The message number has the format yxxx, where y specifies the message's format and xxx uniquely identifies the message within the message format.

Although not shown here, most of the messages are preceded by a carriage return, a line feed, and the date and time. The format of the date and time is as follows:

ddd hh:mm

where:

ddd is the Julian day.

hh is the hour.

mm indicates the minute.

The file DNCOMOBJ.TSKSRC.CCMSG contains templates for these messages. For messages that do not appear in this appendix, refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual.*

The messages and explanations are as follows:

1001 INVALID COMM DEVICE SPECIFIED

The communications device specified is incompatible with the Emulator task being activated. Review the responses entered during the execution of the XSGU command (during system generation) that specify protocol types. Refer to the *Model 990 Computer DNOS 3780/2780 Emulator Object Installation Guide* for related information.

1002 BELL RECEIVED

A BELL message has been received from the remote system. The remote station operator requests voice communication on the data line.

1003 BID ERROR

The remote system is not responding to the line bid message sent. Determine if the remote system is active or if it has been disconnected. If it is disconnected, reconnect the link and retry the transmission. If it is active and connected, check the switch settings on the modem, and, if applicable, check the modem options. (See Appendix A for a list of recommended and required options for various modems.)

1004 COMM DEVICE INOPERABLE

The device specified as the communications device name does not work. Ensure that the communications board interrupt is enabled and that the communications device DSR has been generated properly. Check the communications board for possible malfunction.

1005 COMM DEVICE NOT AVAILABLE

The device specified as the communication device name does not exist. Reenter a valid communication device name.

1006 INVALID PROC USED TO BID COMM TASK

A previous version of the Emulator activation procedures (X3780 or X2780) is being used. Ensure that the activation procedure being used is the current version.

1007 COMM COMMAND DEVICE NOT AVAILABLE

The device (or file) specified as the command source access name cannot be opened. Check to see if the device (or file) is already open or if it exists.

1008 COMMAND/LOG FILE ASSIGN/OPEN ERROR

An error has occurred in attempting to open or to assign a logical unit number to a command file or a log file. Ensure that the pathname is correct, that the device or file is not already open, and that the device (if specified) exists. If these prerequisites are satisfied, reenter the command.

1009 CONNECT FAILED

A call, attempted the number of times specified in the installation procedure (AUTO CALL RETRY COUNT), has failed. The called number is busy, a wrong number was dialed, or no answerback tone was received from the called modem. Reenter the command to establish the desired connection.

1010 CONNECT SUCCESSFUL

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

1011 IDLE ERROR

No data has been transferred over the communications link for the idle line time-out interval specified during installation. During installation, the operator specified the parameter Disconnect on Error, and the IDLE Line Threshold parameter has been exceeded. The line is disconnected.

1012 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 and is connected or has a connection in process, the system issues this message and halts further command input until buffer space becomes available through the processing of commands; it then continues. If the system is in the operator mode or disconnected, the system issues this message, aborts the last command and then issues a new COMMAND: prompt. The number of pending receive and send requests specified during installation defines the buffer space reserved for this purpose. (See Appendix B.)

1013 ID xxxxx

This message identifies the transaction ID number assigned to the last transaction send or receive command and acknowledges that it has been queued on the send or receive queue.

1014 INVALID PARAMETERS

A response which exceeds the maximum length for a valid pathname (48 characters) has been made to one of the activation prompts. Examine the responses entered and make the necessary correction.

1015 LINE DISCONNECTED

The communication link has been successfully disconnected as requested (using a disconnect command or disconnect on error option specified during installation), or a disconnect that was not requested has been sensed (that is, a DLE EOT message has been received or the modem's data set ready (DSR) condition has been dropped).

1016 COMM LOG DEVICE NOT AVAILABLE

The device or file specified as the log output cannot be opened. Check to see if the device (or file) is already open or if the device, if specified, exists.

1017 NO RECEIVE PATHNAME SPECIFIED

The receive pathname recognition code has been received and the receive pathname option is on, but no pathname is specified. This can be corrected only at the remote station. The receive pathname recognition code syntax requires correction, after which the remote station can reattempt transmission.

1019 SEND REQUEST COMMAND ERROR

A send request was received that could not be processed correctly due to an illegal pathname or insufficient command buffers. The send request is ignored. Correct the pathname syntax or wait for an outstanding send or receive request to be completed before reissuing the send request.

1020 RPNAME COMMAND ERROR

A receive pathname specification string has been received but could not be processed correctly due to an illegal pathname or insufficient command buffers. The specification is ignored. Correct the pathname syntax or wait for the completion of an outstanding send or receive request before reissuing the request.

1021 TRANSACTION IN PROGRESS

An ABORT option specified a transaction ID on the send or receive queue for which a data transfer was in progress. The ABORT request is ignored. Abort the transaction using the ABORT*COM command, which will disconnect the line. Alternatively, request that the remote end issue a RESET command, which will send an EOT after the next block is received.

1022 TRANSACTION NOT ON QUEUE

An ABORT option specified a transaction ID that was not present on the send or receive queue. The ABORT request is ignored. Display the transaction queues, using the ST*ALLQ command, to locate the specified transaction identifier.

1023 TRANSLATION TABLE LOAD ERROR

The character translation table overlay (ASCII/EBCDIC, EBCDIC/ASCII) could not be loaded into memory. The Emulator task terminates. Examine the program file to see if it is intact and if the translation table overlays are present in the program file (overlays >F0 through >FF).

1024 LOG QUEUE FULL

More than eight operator messages are queued in the task buffers for display on the operator console or log. The Emulator halts message output until the queue empties. This condition should not occur unless the operator has disabled the log device and delayed restarting the output.

1025 COMM JOURNAL FILE NOT AVAILABLE

The file specified for journal output cannot be opened. Check to see if the device or file is already open, or, if a device, verify that it exists.

1026 INVALID JOURNAL/LOG ACCESS NAMES

The response to the LOG ACCESS NAME: prompt during Emulator activation is the same as the response to the JOURNAL ACCESS NAME: prompt.

2001 I/O ERROR xx pathname

During a transmission or reception, an error occurred during input from or output to the specified pathname. The transmission or reception is terminated. The term xx is a system error code. Refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual* for an explanation of the code. The error codes described are listed as internal message codes and are printed as >00xx.

2002 OPEN/POSITION ERROR xx pathname

During a transmission or reception, the Emulator encountered an error while opening or positioning to a record in the specified pathname. The variable xx is a system error code. Refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual* for an explanation of the code. The error codes described are listed as internal message codes and are printed as >00xx.

Check to see if the device or file is already open or, if a device, that it exists. Also, check that the record exists within the file.

3001 RECEIVED SEND REQUEST COMMAND pathname

A send request has been received. The specified pathname is placed on the send queue for subsequent transmission.

3002 RPNAME COMMAND pathname

A receive pathname specification string has been received and placed on the queue for the output device (RPR or RPU).

4001 RPRINT COMPLETE zz iiiii rrrrr ooooo pathname

or

4002 RPUNCH COMPLETE zz iiiii rrrrr ooooo pathname

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

zz is the communication completion/error code. If the completion/error code is not 00, an error has occurred during data reception. Refer either to Table D-1 for a list of the completion/error codes and their meanings or to the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as internal message codes and are printed as >00zz.

iiiii is the receive pathname of the queue entry just completed.

rrrrr is the number of records received by the buffer. The records are then written to the access name currently active in the transaction queue.

ooooo is the number of records received with lengths greater than the receive record length (RPURL or RPRRL) specified.

4003 SEND COMPLETE zz iiiii rrrrr ooooo pathname/text string

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

zz	is the communication completion/error code. If the completion/error code is not 00, an error has occurred and the remainder of the transaction is terminated. Refer either to Table D-1 for a list of the completion/error codes and their meanings or to the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as internal message codes and are printed as >00zz.
iiiii	is the transaction ID of the queue entry just completed.
rrrrr	is the number of records read from the access name currently active in the send queue (that is, waiting for transmission).
ooooo	is the number of records sent with lengths greater than the send record length (SRL) specified.

5001 TASK BID REQUEST RECEIVED xx

A bid task request has been received. The variable xx gives the completion/error code for bidding the task. This code can be a 59, indicating that the syntax of the bid task request is not as described in paragraph 4.5.3.5 or it can be an SVC error code. The SVC error codes are listed in the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as I/O errors and are printed as >00zz.

5002 CALL RETRY NECESSARY xx

An attempt to establish a connection using either a CALL or ANS command has failed (because of a busy signal, wrong number, etc.). If an ACU is being used, another attempt to place the call will be made. The number of times the ACU will retry a call is specified during the installation procedure. Redial the number if it is a manually placed call.

The variable xx is a completion/error code; refer either to Table D-1 for a list of the error codes and their meanings or to the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as internal message codes and are printed as >00xx.

5003 COMM DEVICE NOT AVAILABLE xx

The Emulator has been generated with an incorrect address for the communication device, the communication device is not working, or it is busy. The variable xx is a completion/error code; refer either to Table D-1 for a list of error codes and their meanings or to the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as internal message codes and are printed as >00xx.

5004 STATIC BUFFER ALLOCATION FAILED xx

An attempt to increase the length of the static buffer table area (BTA) has exceeded the system maximum. The number of beets used to increase this area is calculated by the following formula:

$$\frac{\text{Maximum Block Size} + \text{Overhead}}{8}$$

where the maximum block size is the communication block size specified at installation, and the overhead is a constant 49 bytes.

The term xx is a system error code. Refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual* for an explanation of the code. The error codes described are listed as internal message codes and are printed as >4Azz.

NOTE

In most cases, this is a warning message from the Emulator. The Emulator may not be functioning properly depending on the actual amount of BTA used. DNOS system utilities are available to increase the BTA maximum whenever necessary.

5005 TASK BID PROCESSED xx

If xx is nonzero, an error was encountered during the processing of a BID command (see paragraph 4.5.7.4). The completion/error code xx is described in the *Model 990 Computer DNOS Messages and Codes Reference Manual*. The error codes described are listed as I/O errors and are printed as >20xx or >31xx or they are listed in Table D-1. A value of 59 for xx indicates that a syntax problem was encountered in processing the responses to the BID command prompts.

5006 ACU NOT AVAILABLE xx

A CALL command and phone number have been entered on a system with an ACU configured, but the ACU is not available. The CRU address for the ACU defined in the installation procedure may be incorrect, or the phone may be off the hook. The completion/error code xx is listed in the *Model 990 Computer DNOS Messages and Codes Reference Manual* and in Table D-1. For this message, xx can have a value of >AA, >AB, >AC, >AD, >AE, or >AF. Correct the error and dial manually, or issue a DISC command and reenter the CALL command.

6001 TRANSPARENCY ERROR rrrrr pathname

The Emulator has detected an invalid data link control character in the data of a non-transparent transmission. (See Appendix E.) Only the first pathname of multiple pathnames is printed.

rrrrr is the decimal number of the record where the illegal character was found.

7001 message

This identifies an entry in the log and/or journal as an operator message.

message	is the content of the operator message, and is a string from 1 to 48 characters long.

8001 RPUNCH INITIATED iiiii pathname

A punch reception has been initiated.

iiiii	is the transaction ID of the queue entry just completed.
pathname	is the pathname of the file received.

8002 RPRINT INITIATED iiiii pathname

A print reception has been initiated.

iiiii	is the transaction ID of the queue entry just completed.
pathname	is the pathname of the file received.

8003 SEND INITIATED iiiii {pathname/<text string>}

A transmission of data from the indicated pathname or text string has been initiated.

iiiii	is the transaction ID of the queue entry just completed.
pathname	is the pathname for the first file sent.
text string	is the first text string sent.

*** INVALID INPUT

The operator input just entered is incorrect. Check the spelling, or see if the input is valid in context, and reenter. Also, check whether the pathname is too long or contains invalid characters.

** 3780/2780 WRITE ERROR TO LOG ERR = zz **

The Emulator has a message for output to the Emulator log file but cannot access it. The above message is output to the system log. The error code zz appears in the *Model 990 Computer DNOS Messages and Codes Reference Manual.* The error codes described are listed as internal message codes and are printed as >00zz.

** 3780/2780 WRITE ERROR TO JOURNAL ERR = zz **

The Emulator has a message for output to the Emulator journal but cannot access it. The above message is output to the system log. The error code zz appears in the *Model 990 Computer DNOS Messages and Codes Reference Manual.* The error codes described are listed as internal message codes and are printed as >00zz.

The completion/error codes in Table D-1 indicate the status of the Emulator when the message is displayed. For codes not included in this table, refer to the *Model 990 Computer DNOS Messages and Codes Reference Manual.*

Table D-1. Communication Completion/Error Codes

Code	Brief Description	Recovery Information
00	No error	No recovery required.
06	I/O aborted	The Emulator has issued an abort I/O sequence on a transaction in progress. Reissue the Emulator command to retry the operation.
18	Line disconnected	Connection must be reestablished if additional data transfers are desired.
51	I/O terminated abnormally	The Emulator received an EOT prior to receiving an ETX. The data file must be retransmitted to obtain a complete data transfer (see Appendix E for a description of the EOT and ETX characters).
55	RVI error	The Emulator has received an RVI. The Emulator terminates its transmission attempt and enters the idle state to allow the remote station to transmit.
56	NAK threshold exceeded	The communication link is unusually noisy or one of the modems is faulty. Try to reconnect the link to see if another circuit will allow transfers. (See Appendix E for a description of the NAK character.)
57	ENQ threshold exceeded	The remote system is responding incorrectly (wrong acknowledgement) or is not responding at all. Determine if the remote system has disconnected. (See Appendix E for a description of the ENQ character.)
59	Syntax error exists	A syntax error was encountered during the processing of responses to a BID command prompt. Correct the syntax error and reattempt.

Table D-1. Communication Completion/Error Codes (Continued)

Code	Brief Description	Recovery Information
60	Data threshold exceeded	The next character in a block has not been received within the time specified during installation for the RECEIVED DATA: prompt. Determine if the sending station has disconnected.
AA	ACR/DSS not reported	The ACU did not return the ACR or DSS signal within the expected time interval. Reattempt the call. If it is unsuccessful again, check the ACU hardware.
AB	DSR signal not returned	The call was completed and the ACU reported DSS, but the data set ready signal was not returned from the modem within the expected time interval. Reattempt the call. If it is unsuccessful again, check the ACU, modem, and/or communications hardware.
AC	Abort call and retry	During an ACU call attempt, the ACU returned an Abort Call and Retry (ACR) signal. This can occur if, after a time-out (determined by hardware), the call is not answered or is answered by a station that is not a data terminal.
AD	No power on ACU	The ACU power fail signal is true. Check the power connections to the ACU.
AE	Commmunications line occupied	The communications line to the remote station is busy. Either an all-trunks-busy (ATB) condition exists or the modem at the remote station is already in use. Reattempt the call.
AF	ACU not configured in system	Either no ACU exists in the system or the ACU CRU address was set to the default value (>FFFF) during installation. Verify the hardware installation and/or the ACU CRU address.

Appendix E

Binary Synchronous Communications Procedures

E.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 procedures provide a standard framework for the communication exchange, its checking, and its correction. Since this framework is constructed from special data link control characters that the Emulator adds and removes automatically, the files contain only the data to be transmitted. This appendix describes the control characters that the IBM 3780 and 2780 and the Emulator support. This appendix also briefly describes the circumstances in which the characters are employed.

The data stream also includes characters other than the data link control characters as follows:

- The PAD character that follows each transmission. It ensures that the last character of the transmission block is sent prior to the transmitting station turning off. The PAD character consists of eight 1 bits (>FF).

- The cyclic redundancy check (CRC) character. This is also referred to as the block-check (BCC) character. The Emulator detects transmission errors through a CRC performed on the data. Prior to transmitting, the transmitting station accumulates the check character and includes it in the transmission. The numeric binary value of the data in a block following the STX character (except SYN but including ETB and ETX) is divided by a polynomial that is standard to BSC procedures. 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 CRC character and is inserted in the data stream following an ITB, ETB, or ETX character. The receiving terminal performs a similar division upon reception of the data. If the receiving terminal's remainder matches the transmitting terminal's remainder, the transmission block is accepted as error-free.

E.2 DATA LINK CONTROL CHARACTERS

The control characters recognized by the Emulator are as follows:

Character	Meaning
SYN	Synchronous idle
STX	Start of text
ITB (IUS)	End of intermediate transmission block
IRS (RS)	End of record
ETB	End of transmission block
ETX	End of text
EOT	End of transmission
ENQ	Enquiry
DLE	Data link escape
ACK0/1	Alternating affirmative acknowledgement
NAK	Negative acknowledgement
WACK	Wait before transmitting positive acknowledgement
RVI	Reverse interrupt
TTD	Temporary text delay

E.2.1 SYN — Synchronous Idle

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

E.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.

E.2.3 ITB (IUS) — End of Intermediate Transmission Block

The ITB (IUS) character is used to divide a long file transfer into convenient block sizes for error-checking purposes. It does not indicate the end of the transmission or a line turnaround. The ITB (IUS) character is always followed by the CRC characters. The ITB (IUS) character is transmitted only by the 2780 Emulator. Transparent data containing ITBs (IUSs) must be transmitted in fixed-length records for the Emulator to receive it correctly.

E.2.4 IRS (RS) — End of Record

This character indicates the end of a record within a block. The IRS code causes a new line (NL) function without terminating a horizontal tab (HT) format message. Refer to Table 3-2.

E.2.5 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 characters for the block follow the ETB character. The ETB character forces a line turnaround and requires a response from the receiving station.

E.2.6 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 characters for the block follow the ETX character. The ETX character forces a line turnaround and requires a response from the receiving station.

E.2.7 EOT — End of Transmission

This character terminates a transmission and indicates to the receiving station that it should return to the control mode. An EOT also indicates certain error conditions. The DLE EOT sequence on a switched circuit indicates to the receiver that the line should be disconnected.

E.2.8 ENQ — Enquiry

The ENQ character bids 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 can begin. If a NAK response is returned, the requesting station cannot transmit and must repeat its enquiry sequence.

The Emulator uses the ENQ character to request the retransmission of a reply when the anticipated reply is not received. Also, the Emulator uses the ENQ character in the TTD sequence.

E.2.9 DLE — Data Link Escape

This character provides supplementary line control characters such as WACK, ACK0/1, and RVI and enables the transmission of control characters in transparent text. The character combination DLE STX initiates transparent text sequences. Return to normal text mode occurs after either a DLE ITB, DLE ETB, or DLE ETX sequence.

E.2.10 ACK0/1 — Alternating Affirmative Acknowledgement

ACK0 is an affirmative acknowledgement for line bids and even-numbered text blocks. It is composed of the following character sequence:

```
DLE >70
```

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

```
DLE /
```

E.2.11 NAK — Negative Acknowledgement

This character indicates 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).

E.2.12 WACK — Wait Before Transmitting Positive Acknowledgement

A receiving station sends this character as a positive acknowledgement when it is temporarily unable to receive more data blocks (Figure E-1). The 2780 Emulator cannot transmit the WACK character. WACK is composed of the following character sequence:

```
DLE ,
```

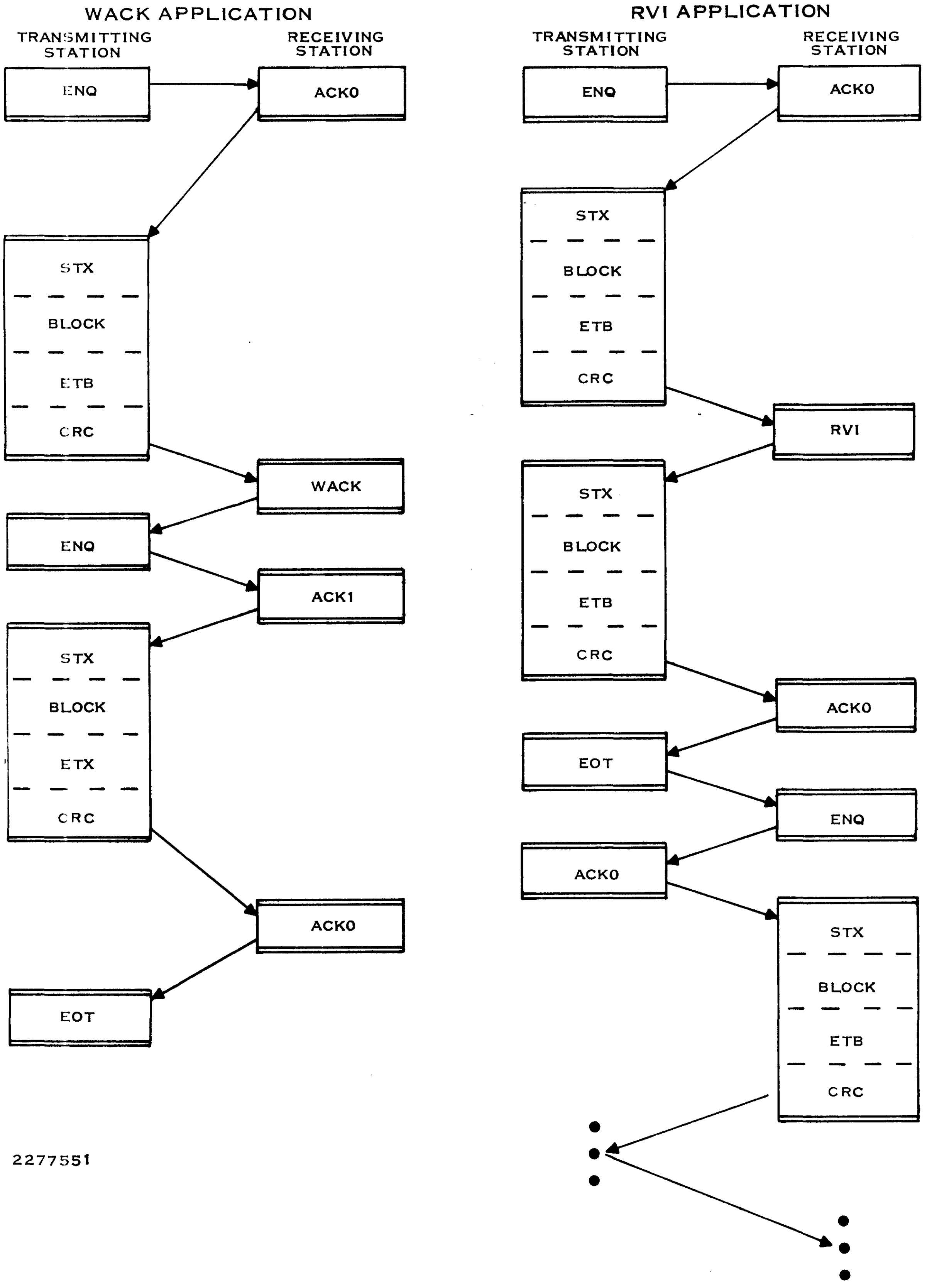


Figure E-1. WACK and RVI Applications

E.2.13 RVI — Reverse Interrupt

A receiving station transmits this character in place of an ACK0 or ACK1 to request that the transmitting station stop sending so that the receiving station can 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 E-1). When the priority data has been transmitted, the original transmitting station again bids for the line and resumes transmitting where it was interrupted. The Emulator cannot transmit the RVI character. RVI is composed of the following character sequence:

DLE @

E.2.14 TTD — Temporary Text Delay

The transmitting station sends this character sequence when the station is not ready to transmit but does not want the line to be disconnected (Figure E-2). The 2780 Emulator cannot transmit the TTD sequence. The TTD sequence is as follows:

SYN SYN STX ENQ PAD

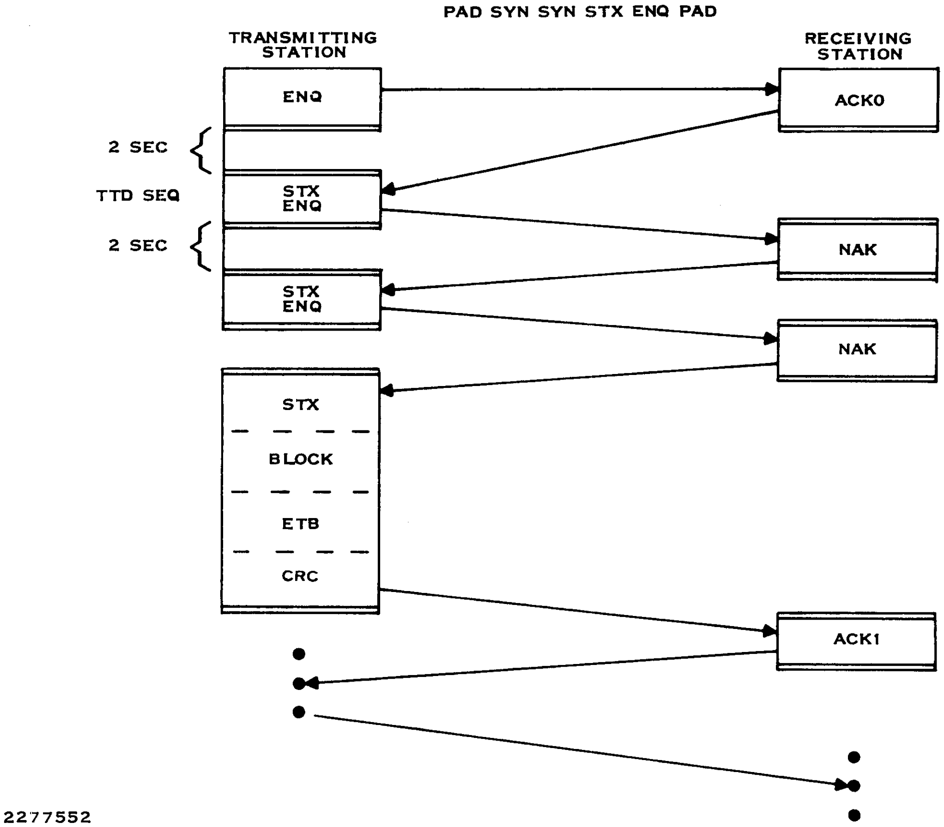


Figure E-2. TTD Application

Appendix F

ASCII/EBCDIC, EBCDIC/ASCII Translation Tables

The IBM 3780 and 2780 protocol employed by the Emulator requires that all data be translated and transmitted as EBCDIC-encoded data unless the binary feature is enabled. With the binary feature enabled, no translation of data occurs and any binary configuration can be transmitted as data, including the full EBCDIC and ASCII character sets.

When the binary feature is disabled, the Emulator automatically translates to EBCDIC user data prepared in ASCII. As a result, 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 greater than or equal to >40 for which ASCII has no equivilent are translated to space characters.

With transparency disabled (TY*OFF), any of the following control characters found embedded in the text during a transmission causes a transparency error:

- For 3780 and 2780 operations:
  - DLE
  - ENQ
  - EOT
  - ETB
  - ETX
  - NAK
  - SOH
  - STX
  - SYN
  - ITB (IUS)

- For 3780 operations only:
  - IRS (RS)
  - IGS
- For 2780 operations only:
  - EM

Also, with transparency disabled, none of the characters DC1, DC2, DC3, and ESC should be transmitted as the first character of a data record; if one of these is transmitted as the first character, the receiving station interprets the character as a device selection or as a VFC specification.

The tables included in this appendix contain the following information:

- ASCII-to-EBCDIC translations; entire ASCII character set (Table F-1).
- EBCDIC-to-ASCII translations; entire EBCDIC character set (Table F-2).
- ASCII-to-EBCDIC and EBCDIC-to-ASCII translations for those characters that differ for the international keyboards supported by Texas Instruments (Table F-3). The country code selected by the operating system determines which translation table is used.
- ASCII-to-EBCDIC translations for Japanese characters (JIS-8) (Table F-4).
- EBCDIC-to-ASCII translations for those Japanese characters that differ from the standard EBCDIC to ASCII translations (Table F-5).

Table F-1. ASCII-to-EBCDIC Translation

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
00	NUL	00	NUL
01	SOH	01	SOH
02	STX	02	STX
03	ETX	03	ETX
04	EOT	37	EOT
05	ENQ	2D	ENQ
06	ACK	2E	ACK
07	BEL	2F	BEL
08	BS	16	BS
09	HT	05	HT
0A	LF	25	LF
0B	VT	0B	VT
0C	FF	0C	FF
0D	CR	0D	CR
0E	SO	0E	SO
0F	SI	0F	SI
10	DLE	10	DLE
11	DC1	11	DC1
12	DC2	12	DC2
13	DC3	13	TM
14	DC4	3C	DC4
15	NAK	3D	NAK
16	SYN	32	SYN
17	ETB	26	ETB
18	CAN	18	CAN
19	EM	19	EM
1A	SUB	3F	SUB
1B	ESC	27	ESC
1C	FS	1C	IFS
1D	GS	1D	IGS
1E	RS	1E	IRS
1F	US	1F	IUS
20	SPACE	40	SPACE
21	!	5A	!
22	”	7F	”
23	#	7B	#
24	$	5B	$
25	%	6C	%
26	&	50	&
27	’	7D	’
28	(	4D	(
29	)	5D	)
2A	*	5C	*
2B	+	4E	+
2C	,	6B	,
2D	–	60	–
2E	.	4B	.
2F	/	61	/
30	0	F0	0
31	1	F1	1
32	2	F2	2
33	3	F3	3
34	4	F4	4
35	5	F5	5
36	6	F6	6
37	7	F7	7
38	8	F8	8
39	9	F9	9
3A	:	7A	:
3B	;	5E	;
3C	<	4C	<
3D	=	7E	=
3E	>	6E	>
3F	?	6F	?

Table F-1. ASCII-to-EBCDIC Translation (Continued)

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC, CHAR
40	@	7C	@
41	A	C1	A
42	B	C2	B
43	C	C3	C
44	D	C4	D
45	E	C5	E
46	F	C6	F
47	G	C7	G
48	H	C8	H
49	I	C9	I
4A	J	D1	J
4B	K	D2	K
4C	L	D3	L
4D	M	D4	M
4E	N	D5	N
4F	O	D6	O
50	P	D7	P
51	Q	D8	Q
52	R	D9	R
53	S	E2	S
54	T	E3	T
55	U	E4	U
56	V	E5	V
57	W	E6	W
58	X	E7	X
59	Y	E8	Y
5A	Z	E9	Z
5B	[	4A	¢
5C	\	E0	\
5D	]	4F	|
5E	∧	5F	¬
5F	–	6D	–

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
60	`	79	`
61	a	81	a
62	b	82	b
63	c	83	c
64	d	84	d
65	e	85	e
66	f	86	f
67	g	87	g
68	h	88	h
69	i	89	i
6A	j	91	j
6B	k	92	k
6C	l	93	l
6D	m	94	m
6E	n	95	n
6F	o	96	o
70	p	97	p
71	q	98	q
72	r	99	r
73	s	A2	s
74	t	A3	t
75	u	A4	u
76	v	A5	v
77	w	A6	w
78	x	A7	x
79	y	A8	y
7A	z	A9	z
7B	{	C0	{
7C	¦	6A	¦
7D	}	D0	}
7E	~	A1	~
7F	DEL	07	DEL

Table F-2. EBCDIC-to-ASCII Translation

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
00	NUL	00	NUL
01	SOH	01	SOH
02	STX	02	STX
03	ETX	03	ETX
04	PF	00	NUL
05	HT	09	HT
06	LC	00	NUL
07	DEL	7F	DEL
08		00	NUL
09	RLF	00	NUL
0A	SMM	00	NUL
0B	VT	0B	VT
0C	FF	0C	FF
0D	CR	0D	CR
0E	SO	0E	SO
0F	SI	0F	SI
10	DLE	10	DLE
11	DC1	11	DC1
12	DC2	12	DC2
13	DC3	13	DC3
14	RES	00	NUL
15	NL	00	NUL
16	BS	08	BS
17	IL	00	NUL
18	CAN	18	CAN
19	EM	19	EM
1A	CC	00	NUL
1B	CU1	00	NUL
1C	IFS	IC	FS
1D	IGS	ID	GS
1E	IRS	1E	RS
1F	IUS	1F	US

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
20	DS	00	NUL
21	SOS	00	NUL
22	FS	00	NUL
23		00	NUL
24	BYP	00	NUL
25	LF	0A	LF
26	ETB	17	ETB
27	ESC	1B	ESC
28		00	NUL
29		00	NUL
2A	SM	00	NUL
2B	CU2	00	NUL
2C		00	NUL
2D	ENQ	05	ENQ
2E	ACK	06	ACK
2F	BEL	07	BEL
30		00	NUL
31		00	NUL
32	SYN	16	SYN
33		00	NUL
34	PN	00	NUL
35	RS	00	NUL
36	UC	00	NUL
37	EOT	04	EOT
38		00	NUL
39		00	NUL
3A		00	NUL
3B	CU3	00	NUL
3C	DC4	14	DC4
3D	NAK	15	NAK
3E		00	NUL
3F	SUB	1A	SUB

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
40	SPACE	20	SPACE
41		20	SPACE
42		20	SPACE
43		20	SPACE
44		20	SPACE
45		20	SPACE
46		20	SPACE
47		20	SPACE
48		20	SPACE
49		20	SPACE
4A	¢	5B	[
4B	.	2E	.
4C	<	3C	<
4D	(	28	(
4E	+	2B	+
4F	|	5D	]
50	&	26	&
51		20	SPACE
52		20	SPACE
53		20	SPACE
54		20	SPACE
55		20	SPACE
56		20	SPACE
57		20	SPACE
58		20	SPACE
59		20	SPACE
5A	!	21	!
5B	$	24	$
5C	*	2A	*
5D	)	29	)
5E	;	3B	;
5F	¬	5E	^

Table F-2. EBCDIC-to-ASCII Translation (Continued)

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
60	–	2D	–
61	/	2F	/
62		20	SPACE
63		20	SPACE
64		20	SPACE
65		20	SPACE
66		20	SPACE
67		20	SPACE
68		20	SPACE
69		20	SPACE
6A	¦	7C	¦
6B	,	2C	,
6C	%	25	%
6D	_	5F	_
6E	>	3E	>
6F	?	3F	?
70		20	SPACE
71		20	SPACE
72		20	SPACE
73		20	SPACE
74		20	SPACE
75		20	SPACE
76		20	SPACE
77		20	SPACE
78		20	SPACE
79	\	60	\
7A	:	3A	:
7B	#	23	#
7C	@	40	@
7D	’	27	’
7E	=	3D	=
7F	”	22	”
80		20	SPACE
81	a	61	a
82	b	62	b
83	c	63	c
84	d	64	d
85	e	65	e
86	f	66	f
87	g	67	g
88	h	68	h
89	i	69	i
8A		20	SPACE
8B		20	SPACE
8C		20	SPACE
8D		20	SPACE
8E		20	SPACE
8F		20	SPACE
90		20	SPACE
91	j	6A	j
92	k	6B	k
93	l	6C	l
94	m	6D	m
95	n	6E	n
96	o	6F	o
97	p	70	p
98	q	71	q
99	r	72	r
9A		20	SPACE
9B		20	SPACE
9C		20	SPACE
9D		20	SPACE
9E		20	SPACE
9F		20	SPACE
A0		20	SPACE
A1	~	7E	~
A2	s	73	s
A3	t	74	t
A4	u	75	u
A5	v	76	v
A6	w	77	w
A7	x	78	x
A8	y	79	y
A9	z	7A	z
AA		20	SPACE
AB		20	SPACE
AC		20	SPACE
AD		20	SPACE
AE		20	SPACE
AF		20	SPACE
B0		20	SPACE
B1		20	SPACE
B2		20	SPACE
B3		20	SPACE
B4		20	SPACE
B5		20	SPACE
B6		20	SPACE
B7		20	SPACE
B8		20	SPACE
B9		20	SPACE
BA		20	SPACE
BB		20	SPACE
BC		20	SPACE
BD		20	SPACE
BE		20	SPACE
BF		20	SPACE

Table F-2. EBCDIC-to-ASCII Translation (Continued)

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
C0	{	7B	{
C1	A	41	A
C2	B	42	B
C3	C	43	C
C4	D	44	D
C5	E	45	E
C6	F	46	F
C7	G	47	G
C8	H	48	H
C9	I	49	I
CA		20	SPACE
CB		20	SPACE
CC	⑀	20	SPACE
CD		20	SPACE
CE	⑂	20	SPACE
CF		20	SPACE
D0	}	7D	}
D1	J	4A	J
D2	K	4B	K
D3	L	4C	L
D4	M	4D	M
D5	N	4E	N
D6	O	4F	O
D7	P	50	P
D8	Q	51	Q
D9	R	52	R
DA		20	SPACE
DB		20	SPACE
DC		20	SPACE
DD		20	SPACE
DE		20	SPACE
DF		20	SPACE

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
E0	\	5C	\
E1		20	SPACE
E2	S	53	S
E3	T	54	T
E4	U	55	U
E5	V	56	V
E6	W	57	W
E7	X	58	X
E8	Y	59	Y
E9	Z	5A	Z
EA		20	SPACE
EB		20	SPACE
EC	⑁	20	SPACE
ED		20	SPACE
EE		20	SPACE
EF		20	SPACE
F0	0	30	0
F1	1	31	1
F2	2	32	2
F3	3	33	3
F4	4	34	4
F5	5	35	5
F6	6	36	6
F7	7	37	7
F8	8	38	8
F9	9	39	9
FA		20	SPACE
FB		20	SPACE
FC		20	SPACE
FD		20	SPACE
FE		20	SPACE
FF		20	SPACE

Table F-3. ASCII-to-EBCDIC and EBCDIC-to-ASCII Translation for International Characters

FRANCE/BELGIUM CHARACTERS

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
21	!	4F	!
23	#	7B	£
40	à	7C	à
5B	o	4A	o
5C	ç	E0	ç
5D	§	5A	§
7B	é	C0	é
7C	ù	6A	ù
7D	è	D0	è
7E	¨	A1	¨

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
4A	o	5B	o
4F	!	21	!
5A	§	5D	§
5F	^	5E	^
6A	ù	7C	ù
7B	£	23	#
7C	à	40	à
A1	¨	7E	¨
C0	é	7B	é
D0	è	7D	è
E0	ç	5C	ç

SWEDEN/FINLAND CHARACTERS

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
21	!	4F	|
23	#	4A	§
24	$	5A	¤
40	É	E0	É
5B	Ä	7B	Ä
5C	Ö	7C	Ö
5D	Å	5B	Å
5E	Ü	5F	¬
60	é	79	é
7B	ä	C0	ä
7C	ö	6A	ö
7D	å	D0	å
7E	ü	A1	ü

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
4A	§	23	#
4F	|	21	!
5A	¤	24	$
5B	Å	5D	Å
5F	¬	5E	Ü
6A	ö	7C	ö
79	é	60	é
7B	Ä	5B	Ä
7C	Ö	5C	Ö
A1	ü	7E	ü
C0	ä	7B	ä
D0	å	7D	å
E0	É	40	É

Table F-3. ASCII-to-EBCDIC and EBCDIC-to-ASCII Translation for International Characters (Continued)

UNITED KINGDOM CHARACTERS

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
21	!	4F	|
23	£	7B	£
5B	[	4A	[
5D	]	5A	]
7E	~	A1	—

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
4A	[	5B	[
4F	|	21	!
5A	]	5D	]
7B	£	23	£
A1	—	7E	~

GERMANY/AUSTRIA CHARACTERS

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
21	!	4F	!
23	#	4A	#
24	$	5A	$
5B	Ä	7B	Ä
5C	Ö	7C	Ö
5D	Ü	5B	Ü
7B	ä	C0	ä
7C	ö	6A	ö
7D	ü	D0	ü
7E	β	A1	β

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
4A	#	23	#
4F	!	21	!
5A	$	24	$
5B	Ü	5D	Ü
6A	ö	7C	ö
7B	Ä	5B	Ä
7C	Ö	5C	Ö
A1	β	7E	β
C0	ä	7B	ä
D0	ü	7D	ü
E0	§	20	SPACE

Table F-3. ASCII-to-EBCDIC and EBCDIC-to-ASCII Translation for International Characters (Continued)

NORWAY/DENMARK CHARACTERS

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
21	!	4F	|
23	#	4A	#
24	$	5A	¤
5B	Æ	7B	Æ
5C	Ø	7C	Ø
5D	Å	5B	Å
7B	æ	C0	æ
7C	ø	6A	ø
7D	å	D0	å
7E	~	A1	ü

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
4A	#	23	#
4F	|	21	!
5A	¤	24	$
5B	Å	5D	Å
6A	ø	7C	ø
7B	Æ	5B	Æ
7C	Ø	5C	Ø
A1	ü	7E	~
C0	æ	7B	æ
D0	å	7D	å
E0	\	20	SPACE

Table F-4. ASCII-to-EBCDIC Translation for Japanese Characters

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
00	NUL	00	NUL
01	TC$_1$ (SOH)	01	SOH
02	TC$_2$ (STX)	02	STX
03	TC$_3$ (ETX)	03	ETX
04	TC$_4$ (EOT)	37	EOT
05	TC$_5$ (ENQ)	2D	ENQ
06	TC$_6$ (ACK)	2E	ACK
07	BEL	2F	BEL
08	FE$_0$ (BS)	16	BS
09	FE$_1$ (HT)	05	HT
0A	FE$_2$ (LF)	25	LF
0B	FE$_3$ (VT)	0B	VT
0C	FE$_4$ (FF)	0C	FF
0D	FE$_5$ (CR)	0D	CR
0E	SO	0E	SO
0F	SI	0F	SI
10	TC$_7$ (DLE)	10	DLE
11	DC$_1$	11	DC1
12	DC$_2$	12	DC2
13	DC$_3$	13	DC3
14	DC$_4$	3C	DC4
15	TC$_8$ (NAK)	3D	NAK
16	TC$_9$ (SYN)	32	SYN
17	TC$_{10}$ (ETB)	26	ETB
18	CAN	18	CAN
19	EM	19	EM
1A	SUB	3F	SUB
1B	ESC	27	ESC
1C	IS$_4$ (FS)	1C	FS
1D	IS$_3$ (GS)	1D	GS
1E	IS$_2$ (RS)	1E	RS
1F	IS$_1$ (US)	1F	US

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
20	(SP)	40	SP
21	!	5A	!
22	"	7F	"
23	#	7B	#
24	$	E0	$
25	%	6C	%
26	&	50	&
27	'	7D	'
28	(	4D	(
29	)	5D	)
2A	*	5C	*
2B	+	4E	+
2C	,	6B	,
2D	–	60	–
2E	.	4B	.
2F	/	61	/
30	0	F0	0
31	1	F1	1
32	2	F2	2
33	3	F3	3
34	4	F4	4
35	5	F5	5
36	6	F6	6
37	7	F7	7
38	8	F8	8
39	9	F9	9
3A	:	7A	:
3B	;	5E	;
3C	<	4C	<
3D	=	7E	=
3E	>	6E	>
3F	?	6F	?

Table F-4. ASCII-to-EBCDIC Translation for Japanese Characters (Continued)

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
40	@	7C	@
41	A	C1	A
42	B	C2	B
43	C	C3	C
44	D	C4	D
45	E	C5	E
46	F	C6	F
47	G	C7	G
48	H	C8	H
49	I	C9	I
4A	J	D1	J
4B	K	D2	K
4C	L	D3	L
4D	M	D4	M
4E	N	D5	N
4F	O	D6	O
50	P	D7	P
51	Q	D8	Q
52	R	D9	R
53	S	E2	S
54	T	E3	T
55	U	E4	U
56	V	E5	V
57	W	E6	W
58	X	E7	X
59	Y	E8	Y
5A	Z	E9	Z
5B	[	4A	£
5C	¥	5B	¥
5D	]	4F	]
5E	∧	5F	¬
5F	—	6D	—
60	`	79	`
61	a	C1	A
62	b	C2	B
63	c	C3	C
64	d	C4	D
65	e	C5	E
66	f	C6	F
67	g	C7	G
68	h	C8	H
69	i	C9	I
6A	j	D1	J
6B	k	D2	K
6C	l	D3	L
6D	m	D4	M
6E	n	D5	N
6F	o	D6	O
70	p	D7	P
71	q	D8	Q
72	r	D9	R
73	s	E2	S
74	t	E3	T
75	u	E4	U
76	v	E5	V
77	w	E6	W
78	x	E7	X
79	y	E8	Y
7A	z	E9	Z
7B	{	C0	{
7C	¦	6A	¦
7D	}	D0	}
7E	~	A1	—
7F	DEL	07	DEL

Table F-4. ASCII-to-EBCDIC Translation for Japanese Characters (Continued)

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
80	SP	40	SP
81	SP	40	SP
82	SP	40	SP
83	SP	40	SP
84	SP	40	SP
85	SP	40	SP
86	SP	40	SP
87	SP	40	SP
88	SP	40	SP
89	SP	40	SP
8A	SP	40	SP
8B	SP	40	SP
8C	SP	40	SP
8D	SP	40	SP
8E	SP	40	SP
8F	SP	40	SP
90	SP	40	SP
91	SP	40	SP
92	SP	40	SP
93	SP	40	SP
94	SP	40	SP
95	SP	40	SP
96	SP	40	SP
97	SP	40	SP
98	SP	40	SP
99	SP	40	SP
9A	SP	40	SP
9B	SP	40	SP
9C	SP	40	SP
9D	SP	40	SP
9E	SP	40	SP
9F	SP	40	SP
A0	SP	40	SP
A1	｡	41	｡
A2	｢	42	｢
A3	｣	43	｣
A4	､	44	､
A5	･	45	･
A6	ｦ	46	ｦ
A7	ｧ	47	ｧ
A8	ｨ	48	ｨ
A9	ｩ	49	ｩ
AA	ｪ	51	ｪ
AB	ｫ	52	ｫ
AC	ｬ	53	ｬ
AD	ｭ	54	ｭ
AE	ｮ	55	ｮ
AF	ｯ	56	ｯ
B0	ｰ	58	ｰ
B1	ｱ	81	ｱ
B2	ｲ	82	ｲ
B3	ｳ	83	ｳ
B4	ｴ	84	ｴ
B5	ｵ	85	ｵ
B6	ｶ	86	ｶ
B7	ｷ	87	ｷ
B8	ｸ	88	ｸ
B9	ｹ	89	ｹ
BA	ｺ	8A	ｺ
BB	ｻ	8C	ｻ
BC	ｼ	8D	ｼ
BD	ｽ	8E	ｽ
BE	ｾ	8F	ｾ
BF	ｿ	90	ｿ

Table F-4. ASCII-to-EBCDIC Translation for Japanese Characters (Continued)

ASCII VALUE	ASCII CHAR	EBCDIC VALUE	EBCDIC CHAR
C0	ﾀ	91	ﾀ
C1	ﾁ	92	ﾁ
C2	ﾂ	93	ﾂ
C3	ﾃ	94	ﾃ
C4	ﾄ	95	ﾄ
C5	ﾅ	96	ﾅ
C6	ﾆ	97	ﾆ
C7	ﾇ	98	ﾇ
C8	ﾈ	99	ﾈ
C9	ﾉ	9A	ﾉ
CA	ﾊ	9D	ﾊ
CB	ﾋ	9E	ﾋ
CC	ﾌ	9F	ﾌ
CD	ﾍ	A2	ﾍ
CE	ﾎ	A3	ﾎ
CF	ﾏ	A4	ﾏ
D0	ﾐ	A5	ﾐ
D1	ﾑ	A6	ﾑ
D2	ﾒ	A7	ﾒ
D3	ﾓ	A8	ﾓ
D4	ﾔ	A9	ﾔ
D5	ﾕ	AA	ﾕ
D6	ﾖ	AC	ﾖ
D7	ﾗ	AD	ﾗ
D8	ﾘ	AE	ﾘ
D9	ﾙ	AF	ﾙ
DA	ﾚ	BA	ﾚ
DB	ﾛ	BB	ﾛ
DC	ﾜ	BC	ﾜ
DD	ﾝ	BD	ﾝ
DE	ﾞ	BE	ﾞ
DF	ﾟ	BF	ﾟ
E0	SP	40	SP
E1	SP	40	SP
E2	SP	40	SP
E3	SP	40	SP
E4	SP	40	SP
E5	SP	40	SP
E6	SP	40	SP
E7	SP	40	SP
E8	SP	40	SP
E9	SP	40	SP
EA	SP	40	SP
EB	SP	40	SP
EC	SP	40	SP
ED	SP	40	SP
EE	SP	40	SP
EF	SP	40	SP
F0	SP	40	SP
F1	SP	40	SP
F2	SP	40	SP
F3	SP	40	SP
F4	SP	40	SP
F5	SP	40	SP
F6	SP	40	SP
F7	SP	40	SP
F8	SP	40	SP
F9	SP	40	SP
FA	SP	40	SP
FB	SP	40	SP
FC	SP	40	SP
FD	SP	40	SP
FE	SP	40	SP
FF	SP	40	SP

Table F-5. EBCDIC-to-ASCII Translation for Japanese Characters

EBCDIC VALUE	EBCDIC CHAR	ASCII VALUE	ASCII CHAR
41	。	A1	。
42	「	A2	「
43	」	A3	」
44	、	A4	、
45	・	A5	・
46	ヲ	A6	ヲ
47	ァ	A7	ァ
48	ィ	A8	ィ
49	ゥ	A9	ゥ
4A	£	5B	£
51	ェ	AA	ェ
52	ォ	AB	ォ
53	ャ	AC	ャ
54	ュ	AD	ュ
55	ョ	AE	ョ
56	ッ	AF	ッ
58	ー	B0	ー
5B	¥	5C	¥
81	ア	B1	ア
82	イ	B2	イ
83	ウ	B3	ウ
84	エ	B4	エ
85	オ	B5	オ
86	カ	B6	カ
87	キ	B7	キ
88	ク	B8	ク
89	ケ	B9	ケ
8A	コ	BA	コ
8B	SP	20	SP
8C	サ	BB	サ
8D	シ	BC	シ
8E	ス	BD	ス
8F	セ	BE	セ
90	ソ	BF	ソ
91	タ	C0	タ
92	チ	C1	チ
93	ツ	C2	ツ
94	テ	C3	テ
95	ト	C4	ト
96	ナ	C5	ナ
97	ニ	C6	ニ
98	ヌ	C7	ヌ
99	ネ	C8	ネ
9A	ノ	C9	ノ
9B	SP	20	SP
9C	SP	20	SP
9D	ハ	CA	ハ
9E	ヒ	CB	ヒ
9F	フ	CC	フ
A0	SP	20	SP
A1	—	7E	—
A2	ヘ	CD	ヘ
A3	ホ	CE	ホ
A4	マ	CF	マ
A5	ミ	D0	ミ
A6	ム	D1	ム
A7	メ	D2	メ
A8	モ	D3	モ
A9	ヤ	D4	ヤ
AA	ユ	D5	ユ
AB	SP	20	SP
AC	ヨ	D6	ヨ
AD	ラ	D7	ラ
AE	リ	D8	リ
AF	ル	D9	ル
B0	SP	20	SP
B1	SP	20	SP
B2	SP	20	SP
B3	SP	20	SP
B4	SP	20	SP
B5	SP	20	SP
B6	SP	20	SP
B7	SP	20	SP
B8	SP	20	SP
B9	SP	20	SP
BA	レ	DA	レ
BB	ロ	DB	ロ
BC	ワ	DC	ワ
BD	ン	DD	ン
BE	゛	DE	゛
BF	゜	DF	゜
E0	$	24	$

Appendix G

Command Summary

Table G-1 lists the commands available to the user of the DNOS 3780/2780 Emulator software package after the desired Emulator has been activated by using the appropriate SCI command. Two other SCI commands (EPF and DPF) installed with the package are available only while the Emulator is deactivated.

Table G-1. Emulator Command Summary

Command	Function	Prompt	Response	Meaning
ABORT	Abort Emulator task	OPTION:	COM	Disconnect link
			TASK	Terminate task
			ALLQ	Clear all queues
			RPR	Clear receive print queue
			RPU	Clear receive punch queue
			S	Clear send queue
			nnnnn	Kill transaction nnnnn
ANS	Answer telephone	none		
BID	Bid local task	PROGRAM:	program name	Task to bid
		STATION:	identifier	Where to execute
			synonym	Where to execute
		EXECUTE:	YES	Concurrent execution
			NO	Suspend Emulator
		PARAMETERS:	parameters	Parameters passed to bid task
BL	Set block length	LENGTH:	block length	Length in bytes
CALL	Make telephone call	NUMBER:	number	Telephone number
CE	Compression/Expansion (3780 Emulator only)	OPTION:	ON	Enable compression
			OFF	Disable compression
CF	Secondary command file	PATHNAME:	pathname	New command file
CLR	Clear flag	NUMBER:	flag number	Flag to clear
DISC	Disconnect line	none		
DISCR	Disconnect remote	none		
EM	End of media (2780 Emulator only)	OPTION:	ON	Enable feature
			OFF	Disable feature

Table G-1. Emulator Command Summary (Continued)

Command	Function	Prompt	Response	Meaning
ID	Identify terminal	SEQUENCE:	identifier	Terminal ID
IF	Execute on condition	OPTION:	CON DISC LID RID S RPR RPU ALLQ EM3780 EM2780 SS SRPR SRPU SREC 0. . .511	Link connected Link disconnected Match local ID Match remote ID Send queue empty Receive print queue empty Receive punch queue empty All queues empty Emulator is 3780 Emulator is 2780 Transmission complete Print reception complete Punch reception complete Transaction complete Flag set
LF	Identify log file	PATHNAME:	pathname	Log destination
RB	Set records per block (2780 Emulator only)	NUMBER:	number	Number of records
RESET	Transmit EOT signal	none		
RPN	Remote receive pathname	OPTION:	ON OFF	Enable feature Disable feature
RPR	Receive printer data	PATHNAME:	pathname	Where received
RPRD	Receive printer default	PATHNAME:	pathname	Default pathname
RPRRL	Receive print record length	LENGTH:	record-length	Length in bytes
RPU	Receive punch data	PATHNAME:	pathname	Where received
RPUD	Receive punch default	PATHNAME:	pathname	Default pathname
RPURL	Receive punch record length	LENGTH:	record-length	Length in bytes
S	Send data	PATHNAME:	pathname <text string>	File sent Text sent
SET	Set flag	NUMBER:	number	Flag to set
SRL	Send record length	LENGTH:	length	Length in bytes
SRQ	Send request	OPTION:	ON OFF	Enable feature Disable feature

Table G-1. Emulator Command Summary (Continued)

Command	Function	Prompt	Response	Meaning
SS	Send data/select device	DEVICE:	PR	Send to printer
			PU	Send to punch
		PATHNAME:	pathname	File sent
			<text string>	Text sent
ST	Statistics	OPTION:	S	Send queue
			RPR	Receive print queue
			RPU	Receive punch queue
			ALL	All statistics
			CT	All since link
			CUR	Current conditions
			ID	Terminal ID
			ALLQ	All queues
TBID	Remote bid task	OPTION:	ON	Enable feature
			OFF	Disable feature
TERM	Terminate Emulator task	none		
TRAILB	Retain trailing blanks	OPTION:	ON	Enable feature
			OFF	Disable feature
TY	Transparency	OPTION:	ON	Enable feature
			OFF	Disable feature
VFC	Vertical format control	OPTION:	ON	Enable feature
			OFF	Disable feature
WAIT	Wait for condition	OPTION:	RPR	Receive print queue empty
			RPU	Receive punch queue empty
			ALLQ	All queues empty
			S	Send queue empty
			CON	Connection complete
			DISC	Link disconnected
$OPR	Operator message	OPTION:	ON	Enable feature
			OFF	Disable feature

Alphabetical Index

Introduction

HOW TO USE INDEX

The index, table of contents, list of illustrations, and list of tables are used in conjunction to obtain the location of the desired subject. Once the subject or topic has been located in the index, use the appropriate paragraph number, figure number, or table number to obtain the corresponding page number from the table of contents, list of illustrations, or list of tables.

INDEX ENTRIES

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 the right side of the listing reference the following manual areas:

- Sections — Reference to Sections of the manual appear as "Sections x" with the symbol x representing any numeric quantity.
- Appendixes — Reference to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.
- Paragraphs — Reference 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 may be 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
- Other entries in the Index — References to other entries in the index preceded by the word "See" followed by the referenced entry.

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer DNOS 3780/2780 Emulator User's Guide (2270520-9701)

Manual Date: 15 April 1982 **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.

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

CUT ALONG LINE

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

TEXAS INSTRUMENTS
INCORPORATED

DIGITAL SYSTEMS GROUP
P.O. BOX 2909 • AUSTIN, TEXAS 78769