60480600

CONTROL DATA

NETWORK PRODUCTS

NETWORK ACCESS METHOD VERSION 1/ COMMUNICATIONS CONTROL PROGRAM VERSION 3 TERMINAL INTERFACES REFERENCE MANUAL

CDC OPERATING SYSTEM:
NOS 2

COMMAND SUMMARY

Command	Page
ct AB = ab [cr]	5-19
ct AR [cr] CR) CR	5-33
ct BF = bf [cr]	5-39
ct BR [= {Y / N}] [cr]	5-24
ct BS = bs [cr]	5-17
BYE [cr]	4-21
ct B1 = b1 [cr]	5-23
ct B2 = b2 [cr]	5-24
ct CI = {ci / CA} [cr]	5-25
ct CH [cr]	5-14
ct CN = cn [cr]	5-18
ct CP [= {Y / N}] [cr]	5-31
ct1 CT = ct2 [cr]	5-17
ct DL = d11 [, d12 , d13] [cr]	5-42
ct EB = {eb / eb, cpr / cpr} [cr]	5-41
ct EL = {el / el, cpr / cpr} [cr]	5-40
END [cr]	4-20, 7-4
ENDL [cr]	7-2
ct EP [= {Y / N}] [cr]	5-48
EX.LOGT [cr]	4-20
ct FA = {Y / N} [cr]	5-35
GOODBYE [cr]	4-21
ct HC = hostname [cr]	4-12
ct HD [cr]	4-12
HELLO [cr]	4-20
ct HN = hn [cr]	4-11
ct HS = hostname [cr]	4-10
IAF [cr]	4-19
ct IC [= {Y / N}] [cr]	5-21

Command	Page
ct IN = {BK / KB / XK / X / PT / XP} [cr]	5-49
ct LK [= {Y / N}] [cr]	4-22
LOGIN [cr]	4-20
LOGOFF [cr]	4-21
LOGOUT [cr]	4-21
ct LI = {li / CA} [cr]	5-26
MCS [cr]	4-20
ct MS = message [cr]	4-22
ct OC [= {Y / N}] [cr]	5-20
ct OP = {DI / PR / PT} [cr]	5-50
ct PA = {E / I / N / O / Z} [cr]	5-46
ct PG [= {Y / N}] [cr]	5-27
ct PL = pl [cr]	5-30
PNI [cr]	4-19
ct PW = pw [cr]	5-29
RBF [cr]	4-19
ct RC [cr]	4-21
ct SE [= {Y / N}] [cr]	5-34
TAF [cr]	4-19
ct TC = {tc1 / tc2 / tc3 / tc4 / tc5} [cr]	5-15
ct TM [cr]	4-21
TVF [cr]	4-19
ct XL = d11 [, d12 , d13 , d14] [cr]	5-44
ct a [cr]	5-23
TEST [cr]	7-1
1 [cr]	7-1
2 [cr]	7-1
3 [cr]	7-1

60480600

CONTROL DATA

NETWORK PRODUCTS

NETWORK ACCESS METHOD VERSION 1/ COMMUNICATIONS CONTROL PROGRAM VERSION 3 TERMINAL INTERFACES REFERENCE MANUAL

CDC OPERATING SYSTEM:
NOS 2

REVISION RECORD

Revision	Description
A (09/01/79)	Original release.
B (02/27/81)	Revised to include 714 support, typeahead, and various technical corrections through PSR level 528.
C (09/30/83)	Revised to reflect the Network Access Method Version 1.6 and the Communications Control Program Version 3.6 with the Network Operating System Version 2 at PSR level 596. The manual's title and type are changed. New text and changes cover multiple-host networks, support of the CDC 721 terminal, block mode terminal operation, HASP preprint and postprint support, 2780 and 3780 support, and X.25 terminal support. This revision obsoletes all previous editions. This manual no longer describes network software operation under NOS Version 1.
D (09/19/84)	Revised at PSR level 617 to document support of the CYBER 170 800 Series models and the CYBER 180 Computer Systems. This is a complete reprint.
E (09/30/85)	Revised to reflect release of NAM Version 1.8 under NOS Version 2.4.2 at PSR level 642. Miscellaneous technical corrections are included.
F (07/31/86)	Revised to reflect release of NAM Version 1.8 under NOS 2.5.1 at PSR level 664. Miscellaneous technical changes are included.

REVISION LETTERS I, O, Q, AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
P. O. Box 3492
SUNNYVALE, CALIFORNIA 94088-3492

or use Comment Sheet in the back of this manual

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision
Front Cover	-
Inside Front Cover	F
Title Page	-
ii	F
iii/iv	F
v	F
vi	F
vii/viii	F
ix	E
x thru xii	F
xiii	C
1-1	C
1-2	C
1-3	E
1-4	E
1-5	C
1-6	E
1-7	E
1-8	D
1-9	E
2-1	C
2-2	E
2-3	E
2-4	C
2-5	D
2-6	E
2-7	D
2-8	C
2-9	E
2-10	E
2-11	F
2-12	E
2-13 thru 2-17	D
2-18 thru 2-22	E
2-23	D
2-24	E
2-25	D
3-1	D
3-2	E
3-3	D
3-4 thru 3-8	C
3-9	D
3-10	E
3-11 thru 3-13	C
3-14 thru 3-26	E
4-1	F
4-2	F
4-3	D
4-4	F
4-5	E
4-6	F
4-7	E
4-8	F
4-9	E
4-10	D
4-11	F
4-12	E
4-13 thru 4-20	F
4-21 thru 4-24	E
5-1	D
5-2	E
5-3	E
5-4	D
5-5 thru 5-11	E
5-12	C
5-13	E
5-14	D
5-15	D
5-16	E
5-17	D
5-18	C
5-19 thru 5-51	E
6-1	E
6-2	D
6-3	C
6-4	C
6-5	E
6-6	C
6-7 thru 6-37	E
7-1	F
7-2	F
7-3	C
7-4	C
A-1	C
A-2	C
A-3	E
A-4	D
A-5 thru A-19	C
A-20 thru A-23	D
A-24	C
A-25	D
A-26	D
A-27 thru A-35	E
A-36 thru A-41	C
A-42	E
A-43	E
A-44	D
A-45 thru A-51	C
A-52	D
A-53	C
B-1	C
B-2 thru B-4	E
C-1 thru C-14	C
C-15	E
D-1	E
D-2 thru D-5	C
D-6	E
Index-1 thru Index-5	F
Comment Sheet/Mailer	F
Back Cover	-

PREFACE

The Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual is an overview of the network products that provide terminal access and remote processing capability. The host software described in this book runs under the Network Operating System (NOS) Version 2 on CONTROL DATA® CYBER 180 Series; CYBER 170 Series; CYBER 70, Model 71, 72, 73, and 74; and 6000 Series Computer Systems. This book also describes the CDC® Communications Control Program (CCP) Version 3.8 for the 255x series of network processing units.

AUDIENCE FOR THIS MANUAL

This manual describes the features and explains the operation of network software from a terminal user's viewpoint. This document does not assume that readers have programming background in network communications. This book does assume that readers are familiar with the hardware operation and possible communication uses of the terminals in which they are interested.

This book covers aspects of network terminal operation that are independent of the host application program providing a service for the terminal user. This document should be used with the reference manual for a terminal service facility program, such as the Remote Batch Facility, the Transaction Facility, or the Interactive Facility. Installations that do not run any CDC-produced application programs can refer to this book for description of those features of the network software available to any terminal user.

ORGANIZATION

The organization of this book is modular. Section 1 summarizes the network structure. General concepts and operations are given in section 2.

Section 3 describes aspects of terminal use or network software use that cannot be generalized; these terminal-dependent topics include notes and cautions. Section 4 describes the minimum actions necessary to use any host software from a terminal and explains the most common interruptions of that use. Section 5 describes features available to more sophisticated users of all terminals.

Section 6 describes the operations of CDC 255x series network processing unit hardware and software for programmers writing terminal emulators or for very sophisticated terminal users who want to take full advantage of all their terminal's features. Section 7 describes the on-line diagnostic host software that can be used by any terminal operator; section 7 can be used without reference to the rest of the book.

RELATED PUBLICATIONS

Related material is contained in the publications listed below; the publications are listed within groupings that indicate relative importance to readers of this manual.

The NOS System Information Manual is an online manual that includes brief descriptions of all NOS and NOS product manuals. To access this manual, log in to NOS and enter the command EXPLAIN.

The following manuals are of primary interest:

Publication	Publication Number
Message Control System Version 1 Reference Manual	60480300
NOS Version 2 Reference Set, Volume 2 Introduction to Interactive Usage	60459660
Network Products Remote Batch Facility Version 1 Reference Manual	60499600
Network Products Transaction Facility Version 1 Reference Manual	60459500

The following manuals are of secondary interest:

Publication	Publication Number
American National Dictionary for Information Processing	FIPS 11-1, ANSI X3/TR-1-77
Additional Controls for Use With ASCII	ANSI X3.64-1977
CCITT Recommendations X.3, X.25, X.28, and X.29	None
Communications Control Program Version 3 Diagnostic Handbook	60471500
COMM 18 Version 2 Operator's Guide	60475410
IBM 2780 Data Transmission Terminal, Component Description	GA27-3005-3†
IBM 3270 Information Display System, Component Description	GA27-2749†
IBM 3780 Data Transmission Terminal, Component Information	GA27-3063-3†
IBM Systems Reference Library General Information -- Binary Synchronous Communications	GA27-3004-2†
Mode 4C Communication Control Procedure	CDC-STD 1.10.020
Network Products Network Access Method Version 1 Host Application Programming Reference Manual	60499500
Network Product Network Access Method Version 1 Network Definition Language Reference Manual	60480000
NOS Version 2 Diagnostic Index	60459390
NOS Version 2 Reference Set, Volume 3 System Commands	60459680
200 User Terminal Operating and Programming Guide	82136000
711-10 CRT Display Terminal Operator's Guide	62034100
713-10 Conversational Display Terminal Operator's Guide	62037900
714-10/20 Remote Terminal Subsystem Operating Guide	82184500
721 Display Terminal Operating Guide/Installation Instructions	62940019
721 Display Terminal Reference Manual	62940020

†Not available from CDC

Publication	Publication Number
731-12 Remote Batch Terminal 200 User Emulation Operating and Programming Guide	82186800
734 Batch Terminal Operator's Guide	62971500
750 Terminal Subsystem Operator's Guide	62951400

Sites within the United States can order CDC manuals from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

Other sites can order CDC manuals by contacting their local sales office.

ANSI Standards can be ordered from Sales Department, American National Standards Institute, 1430 Broadway, New York, New York, 10018.

> This manual describes products intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

If you have access to SOLVER, the CDC online facility for reporting problems, you can use it to submit comments about this manual. When SOLVER prompts you for a product identifier for your report, please specify NA5.

CONTENTS

NOTATIONS xiii

1. INTRODUCTION 1-1

The Network and the System: What and Where Is Each? 1-1
Moving Through the Layers 1-1
Data Communication Layers 1-2
The Terminal Layers 1-2
The Host Layers 1-2
Moving Across the Layers 1-3
Terminals and Communication Lines 1-3
Batch or Interactive? 1-4
Input, Output, or Both? 1-4
Network Processing Units 1-6
Network Host Software 1-6
Network Application Programs 1-7
Network Validation Facility 1-7
Interactive Facility 1-7
Remote Batch Facility 1-7
Transaction Facility 1-9
Terminal Verification Facility 1-9
Message Control System 1-9
PLATO NAM Interface 1-9
Printer Support Utility 1-9
NOS/VE Interactive Facility 1-9
Why Have a Network? 1-9

2. INTERFACE OPERATION 2-1

Information Flow 2-1
Information Structures 2-1
Physical Structures and Network Blocks 2-1
Logical Structures and Physical Blocks 2-1
Network Data Blocks 2-1
Terminal Transmission Blocks 2-1
Upline and Downline Block Sizes 2-2
Console Input Concepts 2-2
Network Line Mode Operation 2-2
Network Block Mode Operation 2-3
Physical and Logical Lines 2-3
End-of-Line Indicators 2-6
Multiple Logical Lines in One Message 2-6
End-of-Block Indicators 2-6
Console Output Concepts 2-6
Batch Device Data 2-6
Console Data 2-6
Line Turnaround Convention 2-7
Console Exchange Modes 2-8
Normalized Mode Operation 2-8
Upline Character Sets and Editing Modes 2-9
Downline Character Sets 2-10
Page Width and Page Length 2-11
Transparent Mode Operation 2-11
Using the Interfaces 2-13
Data Input Transmissions 2-13
Transmission Time and Transmission Block Size 2-13
Line Feed and Transmission Keys 2-15
Input Editing Keys 2-16
Data Output Transmissions 2-19
Blocks, Lines, and Messages 2-19
Interrupting and Discarding Output 2-19
Paging Operations and Marking Input 2-21
Data Input and Output Conversions 2-22
Terminal Input Conversions 2-23
NPU Input Conversions 2-24
Normalized Mode 2-24
Transparent Mode 2-24
Host Software Input Conversions 2-24
Host Software Output Conversions 2-24
NPU Output Conversions 2-25
Terminal Output Conversions 2-25

3. TERMINAL OPERATIONS 3-1

Asynchronous Terminals 3-1
Paper Tape Operations 3-1
Paper Tape Reader/Punches 3-1
Preparing Paper Tape Input 3-2
Beginning and Ending Paper Tape Operations 3-3
Creating Paper Tape Output 3-3
Examples of Paper Tape Use 3-4
Terminal Feature Dependencies 3-4
Using Slave Devices 3-7
Using Devices Controlled by Data Codes 3-7
Using 2741 Terminals 3-8
Coping With Terminal Installation Options 3-8
Output Formatting 3-8
Terminals Using X.25 Networks for Access 3-14
Terminal Feature Dependencies 3-14
Using Slave Devices 3-15
Using Devices Controlled by Data Codes 3-15
Using the PAD Service 3-15
Coping With Terminal Installation Options 3-15
Output Formatting 3-15
Mode 4 Terminals 3-16
Mode 4A Switch Settings 3-16
Mode 4A Loading 3-16
731-12 and 732-12 Remote Batch Terminals 3-16
Terminal Feature Dependencies 3-16
Choosing Punched Card Pattern Translations 3-17
Positioning the Cursor 3-17
Access and Disconnection Procedures 3-17
Using the Devices 3-17
Output Formatting 3-17
HASP Workstation Terminals 3-18
Workstation Setup Before Access 3-18
Using Stream Numbers 3-19
Terminal Feature Dependencies 3-19
Using IBM Host Features 3-19
Using the Devices 3-19
Output Formatting 3-19
Using Terminal or Network Transparent Mode 3-20
Bisynchronous 2780 or 3780 Terminals 3-20
Terminal Setup Before Access 3-20
Using Punch Selection Codes 3-20
Using 2780 Terminals 3-21
Terminal Feature Dependencies 3-21
Using IBM Host Features 3-22
Using the Devices 3-22

Using Terminal or Network Transparent Mode 3-22
Using End-of-Text and End-of-Transmission 3-22
Coping With Transmission Blocks 3-23
Output Formatting 3-23
3270 Terminals 3-23
Terminal Feature Dependencies 3-23
Auto-Recognition 3-23
Output Formatting 3-23

4. ACCESS AND DISCONNECTION 4-1

Gathering Information 4-2
Setting Up Your Terminal 4-4
Connecting Your Terminal to the Network 4-5
Identifying Your Terminal to the Network 4-7
Procedure for Asynchronous Terminals 4-8
Procedure for HASP Terminals 4-8
Procedure for Mode 4 Terminals 4-10
Procedure for Bisynchronous Terminals 4-10
Procedure for IBM 3270 Terminals 4-10
Selecting a Host 4-10
Selecting Any Host Path 4-10
Selecting a Specific Host Path 4-11
Controlling the Display of Host Paths 4-12
Connecting to the Selected Host 4-12
Host Connection Command 4-12
Identifying Yourself and Your Terminal to the Host 4-12
Automatic Login 4-14
Login Dialog 4-16
Connecting to a Network Application 4-16
Automatic Connection 4-17
Manual Connection 4-17
Abbreviated Login and Application Connection 4-19
Switching to a Different Application 4-19
Direct Switches 4-19
Indirect Switches 4-20
Dialog for Direct and Indirect Switches 4-20
Restarting Host Identification 4-20
Direct Restarts 4-20
Indirect Restarts 4-20
Disconnecting From a Host System 4-21
Disconnecting With Application Program Commands 4-21
Disconnecting With Host System Commands 4-21
Disconnecting With a Network Software Command 4-21
Reconnecting to a Host or Connecting to a Different Host 4-22
Interruptions 4-22
Communicating With the Network Operator 4-22
Receiving Messages 4-22
Sending Messages 4-22
Preventing Messages 4-22
Suspensions of Communication 4-22
Connection Failures 4-23
Application Program Failures 4-23
Disconnection From an Application and/or a Host 4-23
Communication Failures 4-23

5. CHANGING THE INTERFACES 5-1

Using Terminal Interface Program Terminal Definition Commands 5-1
Command Format, Syntax, and Constraints 5-11
General Format and Syntax 5-12
Constraints 5-13
Terminal Definition Command Keywords and Values 5-14
Displaying Command Values in Use 5-14
Displaying Device Characteristics (CH) 5-14
Changing Command Recognition 5-15
Changing Terminal Class (TC) 5-15
Changing Network Control Character (CT) 5-15
Controlling Input or Output Content 5-17
Changing Backspace Character (BS) 5-17
Changing Cancel-Input Character (CN) 5-18
Changing Abort-Output-Block Character (AB) 5-19
Regulating Device Input and Output Flow 5-20
Controlling Output (OC) 5-20
Controlling Input (IC) 5-21
Controlling Application Program Operations 5-22
Sending a Priority Data Character 5-23
Changing the User-Break-1 Character (B1) 5-23
Changing User-Break-2 Character (B2) 5-24
Changing the Functions of the Interactive Interrupt Key (BR) 5-24
Controlling Input or Output Format 5-25
Changing Carriage Return Idle Character Count (CI) 5-25
Changing Line Feed Idle Character Count (LI) 5-26
Changing Page Wait Operation (PG) 5-27
Changing Page Width (PW) 5-29
Changing Page Length (PL) 5-30
Controlling Cursor Positioning (CP) 5-31
Controlling Input or Output Transmissions 5-32
Changing Line Speed or Character Set Use (AR) 5-33
Using Special Editing (SE) 5-34
Using Full-ASCII Editing (FA) 5-35
Controlling End-of-Input Indicators and Delimiters 5-39
Changing the Blocking Factor (BF) 5-39
Defining End-of-Line Processing (EL) 5-40
Defining End-of-Block Processing (EB) 5-41
Changing Single-Message Mode Transparent Mode Delimiters (DL) 5-42
Changing Multiple-Message Mode Transparent Mode Delimiters (XL) 5-44
Controlling Parity and Character Echoing 5-46
Changing Parity Processing (PA) 5-46
Changing Echoplex Operation (EP) 5-48
Identifying Console Mechanisms 5-49
Selecting an Input Mechanism and Transmission Mode (IN) 5-49
Selecting an Output Mechanism (OP) 5-50

6. THE COMMUNICATIONS CONTROL PROGRAM AND THE NETWORK PROCESSING UNIT 6-1

Hardware Environment 6-1
2551 Series Communications Processor 6-1
CYBER Channel Coupler 6-1
Cassette Drive 6-2
NPU Console 6-2
Multiplex Subsystem 6-2
Multiplex Subsystem Operation 6-3
Input Multiplexing 6-3
Input Demultiplexing 6-3
Output Multiplexing 6-3
Output Demultiplexing 6-4
Trunk Multiplexing 6-4
Network Block Handling 6-4
Simplified Input Processing 6-4
Simplified Output Processing 6-5
Data Priorities 6-6
Connection Regulation 6-6

Levels of Logical Link Regulation 6-6
Line Status 6-7
Terminal Interface Programs 6-7
ASYNC TIP 6-7
Protocol Assumptions 6-8
Supported Input and Output Mechanisms 6-8
Terminal Code Sets and Parity 6-9
Initial Connection 6-9
Disconnection 6-10
Input Operations 6-10
Output Operations 6-10
Break Significance 6-11
Input Regulation 6-11
Error Recovery 6-11
X.25 TIP 6-12
Protocol Assumptions 6-12
Supported Input and Output Mechanisms 6-13
Code Sets and Parity 6-13
Initial Connection 6-14
Disconnection 6-14
Input Operations 6-14
Output Operations 6-15
Break and Reset Significance 6-15
Input Regulation 6-15
Error Recovery 6-15
MODE4 TIP 6-16
Protocol Assumptions 6-17
Supported Input and Output Mechanisms 6-17
Terminal Code Sets and Parity 6-19
Initial Connection 6-20
Disconnection 6-20
Data Formatting in Normalized Mode 6-20
Normalized Editing Modes 6-20
Input Regulation 6-20
Error Recovery 6-20
HASP TIP 6-21
Protocol Assumptions 6-22
Supported Input and Output Mechanisms 6-23
Terminal Code Sets and Parity 6-25
Initial Connection 6-26
Disconnection 6-26
Data Formatting in Normalized Mode 6-26
Normalized Editing Modes 6-26
Input Regulation 6-26
Output Regulation 6-27
Error Recovery 6-27
BSC TIP 6-27
Protocol Assumptions 6-28
Supported Input and Output Mechanisms 6-29
Terminal Code Sets and Parity 6-31
Initial Connection 6-32
Disconnection 6-32
Data Formatting in Normalized Mode 6-32
Normalized Editing Modes 6-32
Input Regulation 6-33
Error Recovery 6-33
3270 TIP 6-33
Protocol Assumptions 6-34
Supported Input and Output Mechanisms 6-35
Terminal Code Sets and Parity 6-36
Initial Connection 6-36
Disconnection 6-36
Input Regulation 6-36
Error Recovery 6-36

7. TERMINAL VERIFICATION FACILITY 7-1

Selection and Connection 7-1
Dialog and Commands 7-1
Test Operations 7-2
Loopback Test 7-2
Line Test 7-3
Screen Test 7-3
Disconnection 7-4

APPENDIXES

A Character Data Input, Output, and Central Memory Representation A-1
B Informative and Diagnostic Messages B-1
C Glossary C-1
D Sample Terminal Session D-1

INDEX

FIGURES

1-1 Layers of the Network Structure 1-1
1-2 Software Involved in Terminal Communications 1-3
1-3 Device Classifications by Mechanism Types 1-5
1-4 Possible Echoplexing Locations 1-6
1-5 Access to System Features Via Network Application Programs 1-8
2-1 Physical and Logical Information Structures in Normalized Data 2-8
2-2 Physical and Logical Information Structures in Transparent Mode Data 2-13
2-3 Input and Output Code Conversion Points 2-22
3-1 Off-Line Tape for Example 1 3-4
3-2 Example 1 Terminal Session 3-5
3-3 Off-Line Tape for Example 2 3-6
3-4 Example 2 Terminal Session 3-6
4-1 Steps in Accessing Host Software 4-1
4-2 Step 1, Gathering Information 4-2
4-3 Interactive Display Console 4-3
4-4 Interactive Printer Console 4-3
4-5 Remote Batch Terminal 4-3
4-6 Step 2, Setting Up Your Terminal 4-4
4-7 Step 3, Connecting to the Network 4-6
4-8 Terminal With Built-in Acoustic Coupler 4-6
4-9 A Separate Acoustic Coupler 4-7
4-10 Data Set Built Into the Terminal 4-7
4-11 Data Sets With Switches and Buttons 4-7
4-12 Step 4, Identifying Your Terminal to the Network 4-8
4-13 Step 5, Selecting a Host Path 4-11
4-14 Host Availability Display 4-13
4-15 Step 6, Connecting to a Host 4-14
4-16 Step 7, Identifying Your Terminal to the Host 4-15
4-17 Sample Login 4-15
4-18 Step 8, Connecting to a Network Application 4-17
4-19 Step 9, Connection Completed 4-18
5-1 Functional Formats of Terminal Definition Commands 5-12
5-2 Display Terminal Characteristics (CH) Command 5-14
5-3 Terminal Class (TC) Command 5-16
5-4 Network Control Character (CT) Command 5-17
5-5 Backspace Character (BS) Command 5-18
5-6 Cancel-Input Character (CN) Command 5-19
5-7 Abort-Output-Block Character (AB) Command 5-20

5-8 Output Control (OC) Command 5-21
5-9 Input Control (IC) Command 5-21
5-10 Priority Data (User-Interrupt Character) Command 5-23
5-11 User-Break-1 Character (B1) Command 5-23
5-12 User-Break-2 Character (B2) Command 5-24
5-13 Break Key Functions (BR) Command 5-25
5-14 Carriage Return Idle Character Count (CI) Command 5-26
5-15 Line Feed Idle Character Count (LI) Command 5-27
5-16 Page Waiting (PW) Command 5-28
5-17 Page Width (PW) Command 5-29
5-18 Page Length (PL) Command 5-31
5-19 Cursor Positioning Response (CP) Command 5-31
5-20 Automatic Recognition (AR) Command 5-33
5-21 Special-Editing Mode (SE) Command 5-34
5-22 Full-ASCII Editing Mode (FA) Command 5-35
5-23 Blocking Factor (BF) Command 5-39
5-24 End-of-Line Delimiter Control (EL) Command 5-40
5-25 End-of-Block Delimiter Control (EB) Command 5-42
5-26 Single-Message Transparent Mode Delimiters (DL) Command 5-43
5-27 Multiple-Message Transparent Mode Delimiters (XL) Command 5-44
5-28 Parity Processing Selection (PA) Command 5-46
5-29 Echoplexing (EP) Command 5-48
5-30 Input Device Description (IN) Command 5-50
5-31 Output Device Description (OP) Command 5-51
6-1 Basic Components of a CDC Network Processing Unit 6-1
6-2 Basic Components of the Multiplex Subsystem 6-2
6-3 Simplified Input Processing 6-4
6-4 Simplified Output Processing 6-5
7-1 TVF Connection Response Banner 7-2
7-2 Display of Selectable Tests 7-2

TABLES

2-1 Terminal Input Control Keys 2-4
2-2 Input Editing Keys 2-16
2-3 Terminal Output Function Keys 2-20
3-1 Format Effector Operations for Asynchronous and X.25 Consoles 3-9
3-2 Format Effector Operations for Mode 4 Consoles 3-18
3-3 Carriage Control Operations for Mode 4 Printers 3-18
3-4 Format Effector Operations for HASP Consoles 3-20
3-5 Carriage Control Operations for HASP Printers 3-21
3-6 Format Effector Operations for 2780 and 3780 Bisynchronous Consoles 3-24
3-7 Carriage Control Operations for 2780 and 3780 Printers 3-24
3-8 Format Effector Operations for the 3270 Bisynchronous Terminal 3-26
3-9 Carriage Control Operations for 3270 Batch Printers 3-26
5-1 Default Terminal Definitions for Terminal Classes 1 Through 8 5-2
5-2 Default Terminal Definitions for Terminal Classes 9 Through 18 5-4
5-3 Parameter Ranges for Terminal Definition Commands, Terminal Classes 1 Through 8 5-6
5-4 Parameter Ranges for Terminal Definition Commands, Terminal Classes 9 Through 18 5-9
5-5 Mode 4 Terminal Page Widths and Lengths 5-31
5-6 Input Character Editing Summary 5-36
5-7 Parity Bit Processing Transmission Mode Dependencies 5-47
7-1 Legal Responses to TVF Command Prompts 7-1

INTRODUCTION 1

Software users regard a terminal as an extension of a monolithic entity they call The System. For most terminal users, this misconception is a harmless one. For some users, the idea of The System can make terminal operation easier; for others, it hides significant features of a terminal network and makes network access a mysterious and frustrating experience. If you are one of the latter, this book was written to dispel some of your confusion.

This section describes the concepts, hardware, and software that constitute a CDC network. As you will learn, there is a difference between the network and the entity often called The System. A CDC network can include many Systems; it also can be connected to other networks, which include other Systems. Sections 2 through 7 of this manual describe functions localized to portions of the CDC network.

THE NETWORK AND THE SYSTEM: WHAT AND WHERE IS EACH?

A CDC network is a many-layered structure. As shown in figure 1-1, the layers of the network include both hardware and software.

MOVING THROUGH THE LAYERS

The terminal hardware you see is the bottom layer of this structure. The software processing your data runs in an element of the top layer of this structure: one of the host computers that connect with the network.

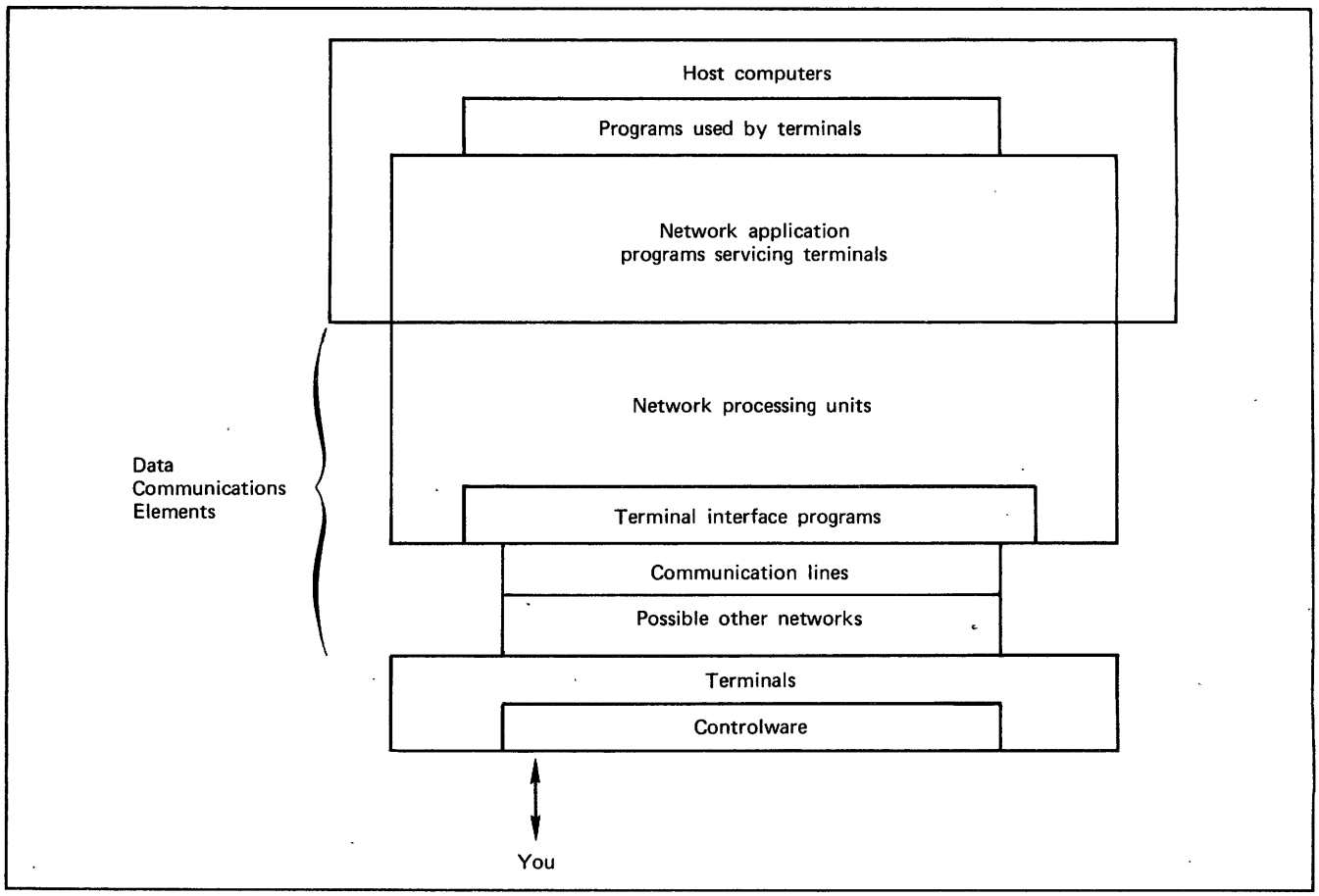


Figure 1-1. Layers of the Network Structure

There are many layers between the top and bottom layers. At the point where two layers adjoin, terminal data is manipulated or routed. Sometimes, you can control these data operations or receive communications describing them.

The top and bottom layers of the network structure are grouped into something called a computer network. When your terminal is connected to a specific host computer, these layers are The System.

CDC network elements are directly controlled by an administrative operator with whom you sometimes communicate. This administrative operator is called the network operator.

The network operator is usually located at the host console. When the network operator is located at a terminal, he or she is called the NPU operator and can perform only the following control function:

> Prevent or permit access of a terminal to the network by enabling or disabling servicing of the terminal by the network. (The network operator can separately control each device and terminal on a communication line, and the line itself.)

If the network operator is located at a host computer console, he or she is called the host operator and can:

> Permit or prevent execution of network application programs.
>
> Shut down executing network application programs.

The remaining layers of the CDC network structure are CDC data communication elements or other communication networks. These layers are not part of The System.

DATA COMMUNICATION LAYERS

CDC data communication elements include both hardware and software. Non-CDC networks are beyond the scope of this manual. If your terminal connects to a CDC network through another network (such as an X.25 protocol packet-switching network or an Ethernet network), you should also read the documentation of that network.

The network operator can indirectly affect your access to The System by controlling the CDC data communication element hardware. He or she can turn the logical paths through the network on or off by enabling or disabling the hardware over which those paths pass. (Paths through the network are described in section 4.)

Software modules important to you run in some of the CDC data communications hardware. These modules are the Terminal Interface Programs.

Terminal Interface Programs (TIPs) run in a network processing unit. One or two network processing units can exist in the data path between the terminal and the host. The TIPs are described in detail in section 6; some of the other NPU software is briefly discussed in that section.

In the layer next to the bottom of the CDC network structure are communication lines. Communication lines are of different types and therefore require different actions to access them and transmit information on them.

The interpretation of information transmitted on a communication line differs according to the rules used by the terminal. The set of rules governing communication with a terminal is called a protocol. Network processing units contain a different Terminal Interface Program for each major protocol supported by the CDC network.

Each layer of the network structure can contain secondary layers. For example, in the bottom layer are the network elements we recognize as terminals. Terminals can be hardware alone, or hardware executing software.

THE TERMINAL LAYERS

Simple terminals (sometimes called dumb terminals) have various controls for the user to enter and manipulate data. More elaborate terminals (sometimes called intelligent terminals) have both hardware controls and software commands for the user.

This second type of terminal includes hardware using an emulator (which can be software, controlware, or permanently programmed hardware called firmware) to imitate the operation of a different kind of terminal. The emulator has its own rules for use, which you must know and which the network cannot know. Section 3 of this manual contains information on some typical emulator requirements; you should read the documentation for the terminal before you try to use section 3.

THE HOST LAYERS

The services you need are grouped into differing types. Each type of servicing is performed by a different terminal servicing facility program, called an application program in many of the network manuals.

Some of these application programs provide access to the host computer operating system so that you can execute programs performing data processing applications. Figure 1-2 summarizes terminal access to this software.

Because there are many sources and destinations for data in the network, the network software includes modules that control routing along the data paths. Some of this software runs in the network processing units. Some of it runs in each host.

The host software is called the network access software in the figure and includes the Network Access Method. The details of Network Access Method use are described in the Host Application Programming reference manual listed in the preface.

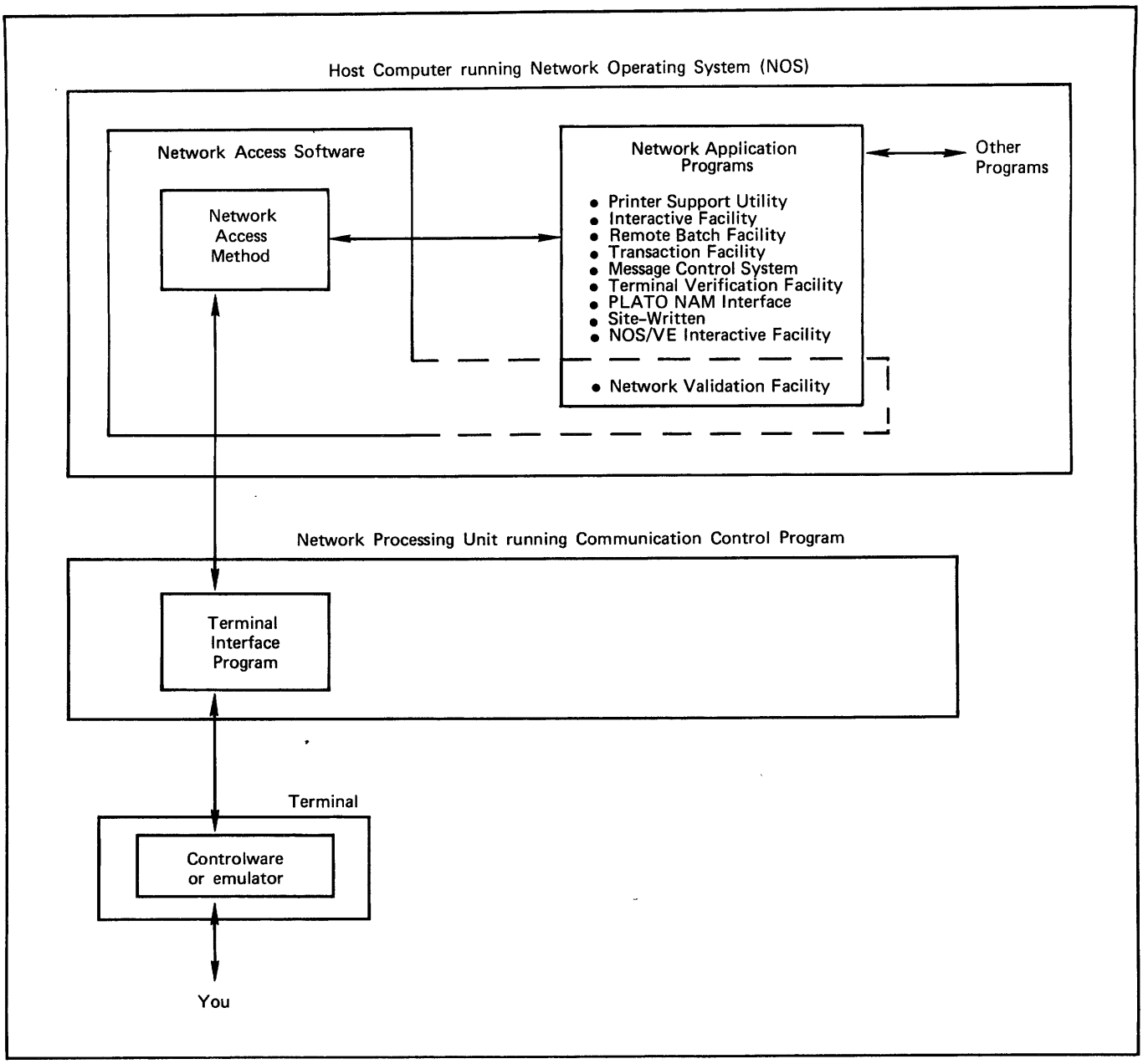


Figure 1-2. Software Involved in Terminal Communications

MOVING ACROSS THE LAYERS

Generally, you have access to all of the layers in the network structure simultaneously. Access to more than one entity within a layer is not possible unless you perform a physical or logical switching operation at a layer closer to you.

For example, the network permits you to switch communication from one terminal servicing facility program (network application program) to another. This switch, however, must be performed using a portion of the network host software known as the Network Validation Facility program. The Network Validation Facility appears in figure 1-2 as part of the network access software.

You can switch from one Terminal Interface Program to another, but only by physically changing the communication line used by the terminal. You can switch the type of communication line protocol used by the terminal, but only when the terminal offers a choice of emulators for different protocols.

TERMINALS AND COMMUNICATION LINES

A CDC network can support terminals using the following communication line protocols:

Asynchronous

Asynchronous through an X.25 packet-switching network packet assembly/disassembly (PAD) service

Synchronous mode 4

Bisynchronous HASP

Bisynchronous 2780, or 3780

Bisynchronous 3270

Each of these basic protocols is subdivided according to the operational characteristics of the terminals that use the protocols.

Asynchronous terminals belong to either:

One of seven teletypewriter-compatible terminal classes using an ASCII code set

One class for terminals compatible with IBM 2741 devices using either an EBCD or a correspondence code set

The teletypewriter-compatible terminal classes can all use communication lines with speeds of 110, 150, 300, 600, 1200, 2400, 4800, or 9600 baud. The 2741-compatible terminal class can use 300 or 134.5 baud communication lines. These classes differ only in the operational characteristics associated with them, as described in sections 5 and 6.

X.25 terminals belong to one of the seven teletypewriter-compatible terminal classes defined for asynchronous terminals using an ASCII code set. These terminals can all use communication lines with speeds up to 19.2 kb/s by accessing the CDC network through another network's PAD service. These terminals differ from asynchronous terminals only in the operational characteristics associated with them, as described in sections 5 and 6.

Synchronous terminals using the CDC mode 4 protocol belong to:

One of two terminal classes using the mode 4A protocol and an ASCII or External BCD code set

One of three mode 4C protocol terminal classes using an ASCII code set

All these mode 4 terminal classes can use communication lines with standard speeds from 2000 to 19.2 kb/s. These classes differ only in the operational characteristics associated with them, as described in sections 5 and 6.

Bisynchronous terminals using the HASP multileaving protocol belong to one of two terminal classes using an EBCDIC code set. One terminal class is for terminals that support preprint carriage control characters, while the other class is for terminals that support only postprint carriage control characters. Terminals of these classes can use communication lines with standard speeds from 2000 b/s to 56 kb/s and have the operational characteristics described in sections 5 and 6.

Bisynchronous terminals using the IBM 2780, 3780, or the 3270 protocol belong to one of three terminal classes using an EBCDIC code set. Terminals of these classes can use communication lines with standard speeds from 2000 b/s to 19.2 kb/s and have the operational characteristics described in sections 5 and 6.

Terminals on synchronous communication lines using the IBM 3270 protocol have the operational characteristics described in section 3.

A CDC network can be modified by an installation to support up to three additional installation-defined protocols, supporting up to four additional installation-defined terminal classes. This manual is restricted to information concerning only protocols and terminal classes of standard CDC networks.

BATCH OR INTERACTIVE?

Your terminal can be serviced as one or more devices by the network software. Figure 1-3 shows how the network communicates with typical terminal devices.

Regardless of the communication line protocol, each terminal has devices that belong in one of two categories. If a terminal device has an input mechanism and an output mechanism that can be used together for two-way dialog, the device is called an interactive device or a console.

If the terminal device has only an input mechanism or only an output mechanism that cannot be used for two-way dialog, the terminal is a batch terminal and the device is a passive or a batch device. All batch terminals must have a console to use the network. IBM 2780 and 3780 terminals often do not have physical consoles; the network pretends that consoles exist at such terminals by allowing console input from a card reader and console output to a line printer.

Batch devices are always operated through an association with a console; this console is called the owning console for the batch device. An associated group of batch devices and their owning console is called a cluster station or a workstation and constitutes one terminal.

INPUT, OUTPUT, OR BOTH?

Because the input mechanism and output mechanism of a console device are physically separate, you can only see your input when something repeats it on the output mechanism. This repetition of input as output is called echoplexing.

Echoplexing can occur at any of several locations, as shown in figure 1-4. Some communication protocols require echoplexing to occur within a terminal. Other protocols allow it to occur outside of the terminal.

External echoplexing normally is performed only when the terminal is set up to use a communication line in a half-duplex mode. Half-duplex operation assumes transmission in only one direction on the line at a time; full-duplex operation assumes simultaneous transmission in both directions. Echoplexing by something outside the terminal normally is not performed when a terminal is set up for full-duplex operation on a communication line.

Echoplexing is the mechanism that lets you verify the entry of data and control characters. For example, pressing a carriage return key on the console keyboard does not automatically cause a carriage return operation on the device output mechanism. The echoplexing of the character code entered by pressing the key is what causes the carriage return operation.

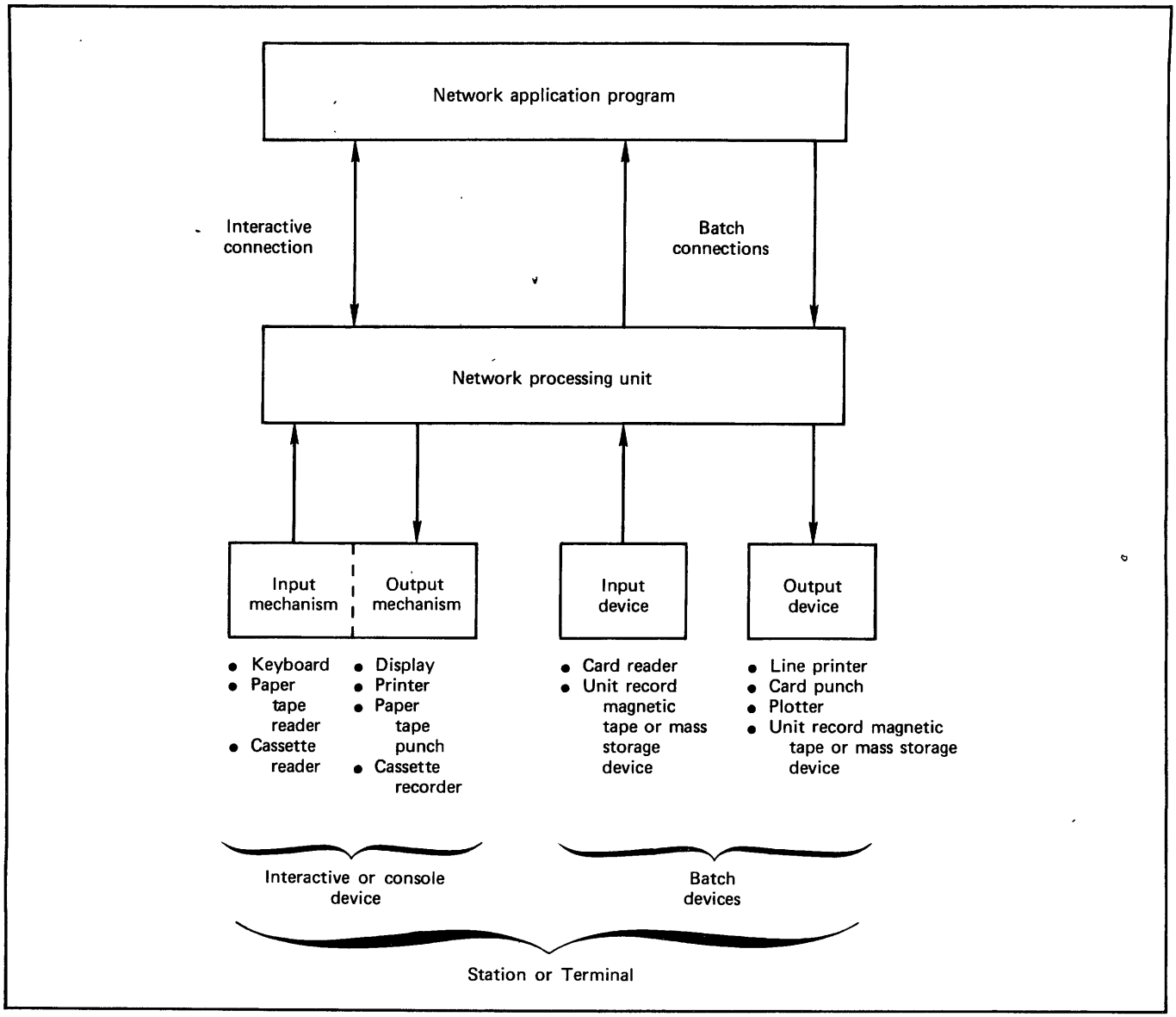


Figure 1-3. Device Classifications by Mechanism Types

The hardware or software echoing the input character back to the output mechanism determines whether or not the character appears on the output mechanism. The farther from the terminal that echoplexing occurs, the longer the delay between the entry of input and its reappearance as output.

Echoplexing by more than one entity in the network garbles the appearance but not the content of any input. On lines used in full-duplex mode, external echoplexing of input during output can garble the appearance and content of the output.

For example, if the network software and your terminal both echoplex your input, the input usually appears twice at your console. If neither the network software nor your terminal echoplex your input, nothing appears at your console. In either case, only one copy of your input reaches the host.

A console can have only one input mechanism and one output mechanism serviced by the network. You can explicitly identify your input mechanism as a keyboard or paper tape reader, and identify your output mechanism as a printer, display, or paper tape punch. If you do not explicitly identify an input mechanism and an output mechanism, the network assumes you are using an appropriate combination.

Consoles can have multiple input mechanisms or output mechanisms on-line at any time, but the network is unaware of all but one. Such mechanisms are only serviced in the sense that they operate in series. Input from a paper tape reader while the keyboard is on-line is treated as keyboard input unless you tell the network otherwise. Output to a hardcopy printer while a display is on-line is treated as display output.

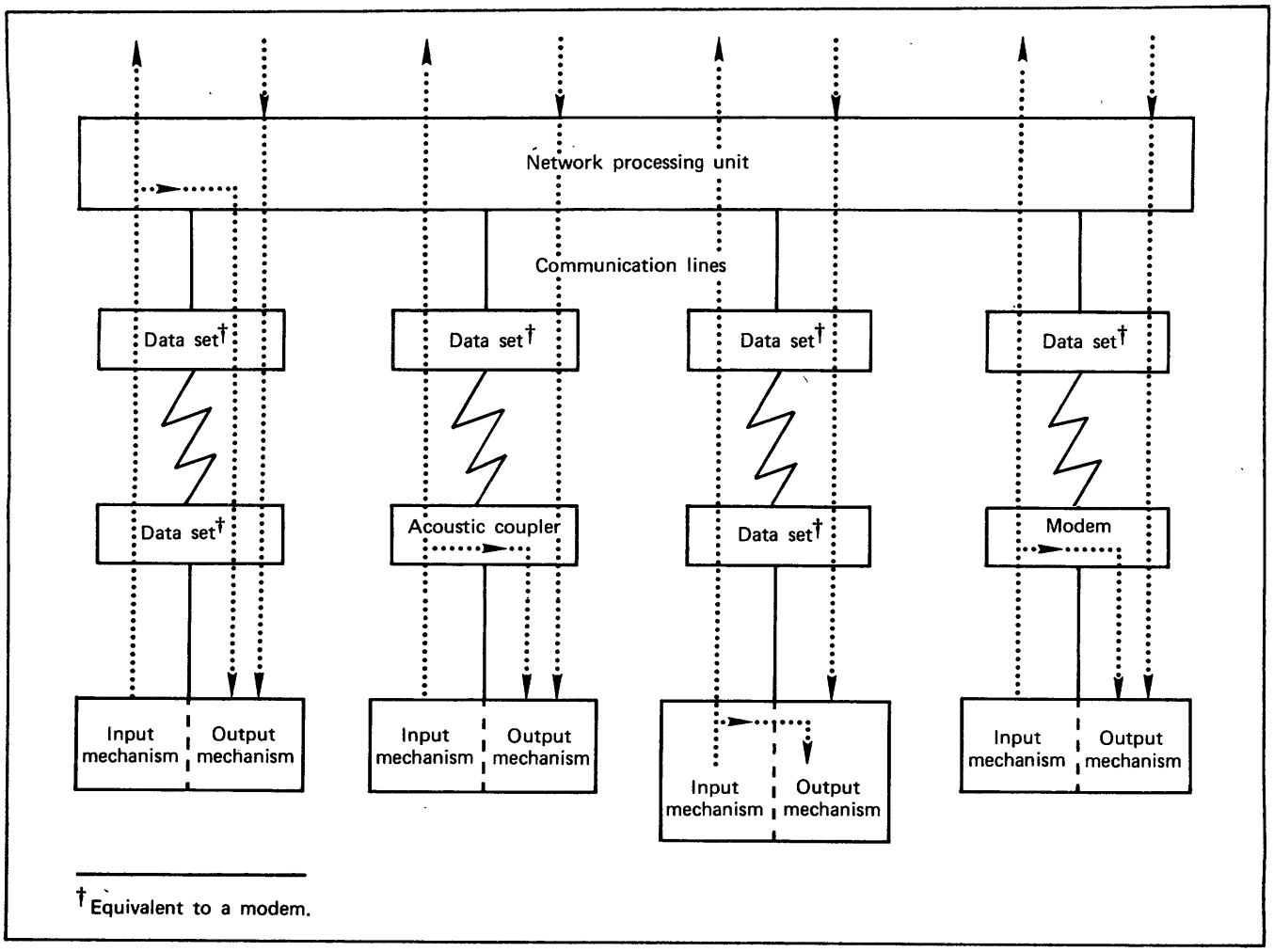


Figure 1-4. Possible Echoplexing Locations

Input or output mechanisms that operate like this are called slaved devices. The most common occurrence of a slaved device is a hardcopy printer that makes a complete record of all output and echoed input while the output and echoed input are being written on a display.

NETWORK PROCESSING UNITS

The network processing units (NPUs) perform functions in addition to those performed by the Terminal Interface Programs. The primary NPU function is the routing and control of data flow through other NPUs, to and from a host computer. This function is usually indistinguishable from the normal buffering of input and output, as described in section 2. Additional detail on data flow through NPUs appears in section 6.

When they route data through the network, the NPUs must perform additional buffering if traffic congestion or temporary communication failures occur. Normally, you cannot detect this buffering. However, sometimes the NPUs run out of buffer space. When this happens, you become aware of the situation through diagnostic messages. These messages are explained in appendix B; the causes and effects are described in section 4.

NETWORK HOST SOFTWARE

As you can see from figure 1-2, network processing units do not communicate directly with the software in the host you are using. Instead, all input and output data processed by an NPU passes through host software explicitly designed to service terminals.

Terminal-servicing software falls into two categories:

Host software that is always used

Host software that is used at your request

The first category of host software is called the network access software. The second category is called network application programs. The relationship between the two categories is shown in figure 1-2. One program, the Network Validation Facility, falls into both categories because of its special supervisory and communication switching roles.

The major portion of the network access software is called the Network Access Method (NAM). Because of the central role played by NAM, many of the operations performed while you use a terminal are performed in conformance with NAM requirements. The Network Access Method is not described directly in this manual; its operation is indirectly described in the context of describing the operation and use of other software.

The operations performed by network application programs are typified by the programs described in the following subsection.

NETWORK APPLICATION PROGRAMS

Network application programs are special programs that provide services to terminal users. Your site can write its own application programs or use CDC-written application programs. CDC programs include:

Remote Batch Facility (RBF)

Interactive Facility (IAF)

Transaction Facility (TAF)

Terminal Verification Facility (TVF)

Network Validation Facility (NVF)

NOS/VE Interactive Facility (VEIAF)

Message Control System (MCS)

PLATO NAM Interface (PNI)

Printer Support Utility (PSU)

Your console can only communicate with one application program at a time. Section 4 of this document describes how you can sequentially access more than one network application program.

Most host computer software can be used through either the Remote Batch Facility or the Interactive Facility. Access to or use of host resources and files via CDC-written application programs is summarized in figure 1-5. The following subsections describe the functions of each CDC-written application program.

Network Validation Facility

The Network Validation Facility (NVF) performs login validation processing. NVF also provides a way to switch your terminal from communicating with one network application program to communicating with another. (You must have a user name valid for access to both programs to accomplish this switch, however; user names are described in section 4.) NVF is the only network application program with which you never need to request communication; a terminal is automatically given as much access to NVF as necessary.

In response to a prompting message from NVF, you enter the name of the network application program to be used next. The name entered must have the form shown in parentheses in the list at the beginning of this subsection. No parenthetical form is shown for NVF because you cannot request the Network Validation Facility directly.

There are many possible forms of the dialog you can have with NVF. The form you use is partially determined by the way the terminal name is configured. The general form of this login dialog is described in each of the reference manuals for network application programs, and in section 4 of this book.

Interactive Facility

IAF comprises six subsystems that permit you to create files and create or execute programs from the console, without use of card readers or line printers. These subsystems are:

BASIC, an interactive version of the interpreter for the BASIC programming language.

FORTRAN, an interactive version of the FORTRAN 5 compiler.

FTNTS, an interactive version of the FORTRAN Extended 4 compiler.

EXECUTE, a quick method to run frequently used programs previously stored in the system in object-code form.

NULL, a quick method to manipulate files and clear subsystem association.

BATCH, which permits you to enter and execute operating system commands one at a time. Any software product for which a command exists can be used through this subsystem; any program can be executed and accessed through this subsystem.

Files you create can be transferred through another computer on your network by using the MFLINK and MFQUEUE NOS commands.

Another subsystem permits communication between terminals using IAF. This subsystem, called ACCESS, is not available to all terminal users.

You can switch your console from one subsystem to another within IAF, from one subsystem to another network application program, or from IAF to NVF.

Remote Batch Facility

RBF permits you to enter a job file from a remotely located card reader and receive job output at remotely located batch devices. You can track the job's progress through the system from the console and manipulate the job's files before and during output. The console can be switched directly from RBF to IAF, or from RBF to NVF (to request connection with another application program).

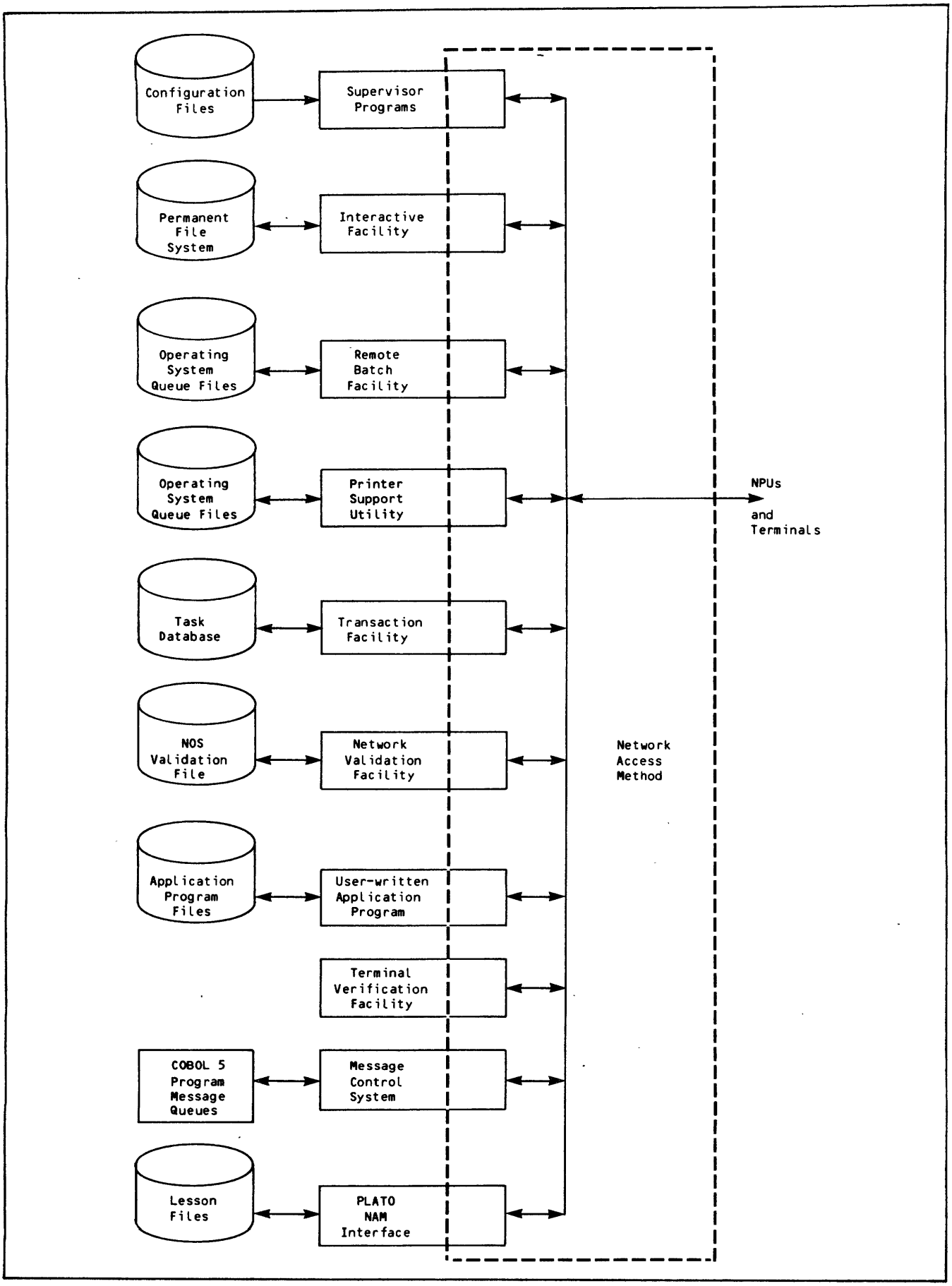


Figure 1-5. Access to System Features Via Network Application Programs

Transaction Facility

TAF permits you to access a database specifically created to treat any operation performed by a console as a transaction that reads or alters that database. TAF use requires very little dialog.

TAF does not permit your console to be switched directly to another application program. The console can be switched from TAF to NVF.

Terminal Verification Facility

TVF provides a selection of tests to verify that the terminal is sending and receiving data correctly. The console can be switched from TVF to NVF. Use of TVF is described in section 7 of this manual.

Message Control System

MCS allows you to queue, route, and journal messages between COBOL programs and terminals. By using the Application Definition Language, an MCS application can be tailored to fit a user's needs. The console can be switched from MCS to NVF.

PLATO NAM Interface

PNI allows operators of CDC 721 Viking Extended terminals to log into the PLATO lesson delivery and authoring software in normal asynchronous mode. The PLATO software then loads its special software into the 721 so that the operator can execute lessons or use the other facilities of the PLATO software. PNI does not allow the console to be switched to another application program.

Printer Support Utility

PSU allows the queuing of printer listings and files to serial (asynchronous) devices. Such devices are considered central site printers; operator control is via the central site's NOS K-Display.

NOS/VE INTERACTIVE FACILITY

VEIAF allows you to access the NOS/VE operating system on CYBER 180's running in dual state.

WHY HAVE A NETWORK?

The preceding pages probably have you wondering why such a complex thing as a network is needed. A network provides your site with many capabilities not readily apparent.

The network allows many more terminals to access a host computer than can be connected directly to the host. The network processing units provide temporary storage of input and output, which lessens the need for storage in the host.

The network allows any application program to service any terminal, regardless of the type of terminal. Without the network, each program would have to separately cope with the details of transmissions to and from particular terminals (details such as storage and echoplexing).

The network allows terminals to share all network resources. Without the CDC network concept, a separate set of terminals would have to be permanently assigned for access to each resource as a dedicated network.

The CDC network concept can easily cope with a terminal that needs to use more than one host. Additional terminal communication protocols can be added without redoing the network application programs.

Finally, and most importantly, the network allows you to use the same software from different terminals without drastic changes in the way your input and output must be handled. Some distinctions dependent on your terminal are inevitable; however, as far as possible, your input and output are always treated as described in the next section of this book.

INTERFACE OPERATION 2

This section describes the way the network handles data from your terminal. Most of the information in this section applies only to consoles.

You can modify some of the operations described in this section by using commands to the Terminal Interface Programs. Those commands are described in section 5.

Before we describe console operation, we must define some terms and concepts. The following subsections will help you understand the remainder of this section and the rest of the manual.

INFORMATION FLOW

Information flow in the network is defined from the viewpoint of the host computer. Information coming to the host is said to be traveling upline; information moving away from the host is said to be traveling downline.

INFORMATION STRUCTURES

The network software uses information structures of two types:

A logical structure based on the concept of a logical line or a message

A physical structure based on various definitions of a block of data

The conditions that create a logical line or message and the conventions governing the subdivision of messages are influenced by the physical structures the network uses. The events involved in actually creating a message are described later in this section under the headings Console Input Concepts and Console Output Concepts.

PHYSICAL STRUCTURES AND NETWORK BLOCKS

Data exchanges between application programs and a terminal occur in logical messages comprising one or more physical network blocks. A network block is a physical subdivision of a logical entity.

A network block is a grouping of information with known and controllable boundary conditions, such as length, completeness of the unit of communication, and so forth. Information exchanges between network processing units and application programs occur in single network blocks.

Information exchanges between network processing units and terminal devices use a different physical structure. Such exchanges occur in sets of character bytes and control bytes called transmission blocks.

LOGICAL STRUCTURES AND PHYSICAL BLOCKS

Upline and downline information within the network is always grouped into physical network blocks. Network data blocks are grouped into logical messages. Messages exchanged between an NPU and a device can also be grouped into physical transmission blocks of one or more logical messages. These concepts are described in more detail later in this section.

Network blocks are restructured into other types of blocks at points of entrance and exit from the network processing units.

NETWORK DATA BLOCKS

A network data block is a collection of character bytes, analogous to a clause in English. It is a partially independent unit of information and might need to be used with other blocks to form a message.

A network data block can contain all or part of a message. Whether a message must be divided into several network data blocks is determined by the size of the message and the size of a network data block.

TERMINAL TRANSMISSION BLOCKS

Terminals send or receive data in physical groupings of character bytes; these groupings are called transmission blocks. The size of a downline transmission block for a specific device is chosen by the network administrator. The value used might be dictated by hardware requirements.

Transmission blocks exchanged with X.25 devices are called packets and have different size and content requirements than transmission blocks exchanged directly with a terminal. The network administrator can control some of the characteristics of packets. If you can change any of your X.25 terminal's packet assembly/disassembly (PAD) service parameters, you can also control some of the characteristics.

During upline transmissions from a console, the NPU reassembles the terminal's transmission block into network blocks. Each transmission block can contain:

One character of a single message

A larger part of a single message

All of a single message

Several messages.

During downline transmissions, the NPU reassembles network blocks into terminal transmission blocks. Each transmission block can contain part of a single message or all of a single message; downline transmission blocks do not contain more than one message.

UPLINE AND DOWNLINE BLOCK SIZES

CDC-defined consoles have network data block sizes that are multiples of 100 character bytes for upline data and of varying sizes for downline data. The last block of an upline message need not contain a multiple of 100 characters.

CDC-defined batch devices have network data block sizes that are multiples of 64 central memory words. Each such block is one mass storage physical record unit (PRU) of a file.

The network administrator initially chooses the appropriate size of upline and downline network data blocks for each device. Sizes are usually chosen to fit a single message into a single network data block, or to optimize use of available network storage, or to meet some other administrative goal.

The initial size of an upline network data block is established by the site administrator when he or she defines the device that produces the block. That size determines the maximum number of characters an application program can receive as a single network data block. When an upline message is too long to fit into a single network data block, the NPU divides it into as many blocks as necessary before delivery to the application program.

You can change the established upline block size using the blocking factor (BF) command described in section 5. You cannot change the established downline block size.

The initial size of a downline network data block is chosen by the site administrator when he or she defines the device that receives the block. The chosen size is a recommended maximum for the number of characters an application program should send in a single network block. The actual maximum size of a downline network block is chosen by the application program sending the block.

The maximum length used for each network data block to or from a device can be independent of the terminal's transmission block size. For example, a mode 4 console cannot accept a transmission block containing more than a specified number of characters.

An application program could divide a multiple line display transmitted to a mode 4 console into network blocks smaller than the buffer space of the terminal. However, the application program does not need to divide its network blocks. The network software reconstructs any of the program's network data blocks longer than the terminal's buffer space into several terminal transmission blocks of the correct size.

The upline block size is enforced by the network software, which subdivides or combines terminal transmission blocks input from a device into network data blocks of that size or smaller. The upline block size defines the largest block that NAM will deliver to the application program.

The downline block sizes defined are advisory values. An application program can accept the size specified for a given connection when the connection is made, or ignore that specification and choose its own value for maximum block size. If an application program transmits blocks larger than the downline block size, the network software does not subdivide them until it creates transmission blocks for the terminal.

CONSOLE INPUT CONCEPTS

A console can send or receive data in two modes:

Normalized mode

Transparent mode

The significance of these data modes is described later in this section under Console Data. The following discussion does not apply to transparent mode data.

In normalized mode, a console transmits logical lines of data. Each logical line is analogous to an English sentence. It is a complete unit of information. In a CDC network, a logical line is a message.

The device can transmit these lines one at a time, with output expected between them; or it can transmit them in sets, with output expected only after the last line of a set. The device therefore can use one of two possible network transmission modes.

If the device transmits only one character or one logical line in each transmission block, it is operating in network line mode. If the device can transmit more than one logical line in a transmission block, it is operating in network block mode.

HASP, bisynchronous devices (terminal classes 9, 14, 16, 17, and 18), always operate in network line mode. Mode 4 devices (terminal classes 10 through 13 and 15) are always considered to operate in network block mode. Only devices in terminal classes 1 through 3 and 5 through 8 can operate in both modes.

NETWORK LINE MODE OPERATION

From your viewpoint, transmitting a single character or one logical line at a time is a buffered line mode form of input. This buffered network line mode allows you to select either terminal character mode or terminal line mode transmission (some devices have switches to select either option) without distinction.

Each logical line is terminated by an end-of-line indicator; this indicator must be transmitted with the line from the terminal. If the terminal buffers lines of input and transmits only when a special key is pressed, the key must either transmit the end-of-line indicator after sending your stored line, or the last character you entered in the line must be the end-of-line indicator and the key should transmit nothing other than the stored line. Each logical line becomes a separate network message when the NPU receives it.

When the NPU is told that an interactive device is operating in network line mode, the NPU reverses the flow of message traffic (performs line turnaround) for it after each upline message. When a message is sent upline in this mode, the NPU begins to send any downline data available for the device. That is, output is allowed after each logical line of input. (Refer to the KB option for the IN command, described in section 5.)

NETWORK BLOCK MODE OPERATION

Some devices can transmit many logical lines in a single transmission block. (The terminal user sometimes can select or override this condition with a BLOCK or BATCH mode switch on the device.) Such devices are called block mode terminals.

Block mode terminals group logical lines in the terminal until the transmission key is pressed; these groups reach the network software as a single transmission block. The network software forwards each message to the application program as a separate transmission; the effect resembles typeahead entries from network line mode terminals.

Each logical line within the input transmission block ends with an end-of-line indicator. Each transmission block is terminated by an end-of-block indicator. The key that transmits the block from the terminal must either transmit the end-of-block indicator after sending your stored lines, or the last character you entered in the last line must be the end-of-block indicator and the key should transmit nothing other than the stored lines.

You or the application program can change the indicators. You or the program can also determine whether each logical line in a transmission block becomes a separate message or each transmission block becomes a single message (refer to the EL and EB options of the EB command, described in section 5).

When the NPU is told a console is operating in network block mode, the NPU does not perform line turnaround for it until all of its current transmission block is received. When the terminal is serviced in this mode, the NPU holds all downline data available for the device until it detects the end-of-block indicator. Output is allowed after each logical line of input only if each logical line of input is transmitted in a separate block. (Refer to the BK and PT options for the IN command, described in section 5.)

A block mode terminal might have a block transmission key that generates a code other than the end-of-block indicator. When the block transmission key generates the end-of-line indicator, the terminal is operating in network line mode, and logical lines are transmitted from the terminal as separate messages.

When the transmission key does not generate either the currently defined end-of-line indicator or the currently defined end-of-block indicator, you must be aware of the distinction. If the key generates another character, you must change the currently defined end-of-block indicator to match that character. If you do not, the character is interpreted as data in an incomplete logical line. If the key generates no code, you must enter an indicator as the last data character before pressing the transmission key. These possible conditions exist:

If the transmission key is pressed immediately after pressing the key that generates an end-of-line indicator, a message is generated. This result is the same as if the device were operating in line mode and the key generating an end-of-line indicator had been pressed.

If the transmission key is pressed immediately after pressing the key that generates an end-of-block indicator, a message is generated. This result is the same as if the device were operating in line mode and the key generating an end-of-line indicator had been pressed, or as if the transmission key had generated an end-of-block indicator.

If the transmission key is pressed without pressing an end-of-line key or end-of-block key as the last prior activity, an incomplete message exists. The Terminal Interface Program (TIP) generates an upline network data block if enough information was received. If downline blocks subsequently become available for the device, the data remains queued while the TIP waits for completion of the input transmission block. This situation exists until you enter more data, ending with either an end-of-line or an end-of-block indicator.

PHYSICAL AND LOGICAL LINES

A logical line of input can contain one or more physical lines; a physical line ends when vertical repositioning of the cursor or carriage occurs. If the device recognizes a line feed operation distinct from a carriage return operation, a physical line ends when a line feed is entered. If no distinction exists between vertical and horizontal repositioning, a physical line is identical to a logical line.

A physical line of input is relevant to the network software when a backspace character is processed. You usually cannot backspace across physical line boundaries to delete characters in physical lines other than the current one.

A physical line ended by a line feed can be separately transmitted to the host as a network block within a longer message, if you want that to occur. This requires using the blocking factor command, discussed in section 5. The keys that indicate a line feed to the network are listed in table 2-1 for typical terminals of each terminal class.

TABLE 2-1. TERMINAL INPUT CONTROL KEYS

Terminal	Terminal Class	Function					
		Terminal-to-NPU Transmission			End-of-Block Indicator (cr)	End-of-Line Indicator (cr)	End of Physical Line (Line Feed)
		Character (1)	Line (1)	Block (2)			
Teletype Model 30 Series Keyboard mode (IN=KB)	1	All	None	None	CTL and D (3)	RETURN	LINE FEED
Teletype Model 30 Series Paper tape mode (IN=PT)	1	None	None	START	X-OFF after RETURN	RETURN	LINE FEED
CDC 713	2	All	None	None	CNTRL and D (3)	RETURN	
CDC 751 or 752 or 756 (CHARacter switch setting)	2	All	None	None	CONTROL and D (3)	CARRIAGE RETURN	LINE FEED
CDC 751 (LINE switch setting)	2	None	CARRIAGE RETURN	None	CONTROL and D (3)	CARRIAGE RETURN	LINE FEED
CDC 751 or 756 (BLOCK switch setting)	2	None	None	SEND	CONTROL and D (3)	CARRIAGE RETURN	LINE FEED
CDC 751 (cassette player)	2	None	None	READ TAPE	CONTROL and S (3)	CARRIAGE RETURN	LINE FEED
CDC 721 Viking Extended (LARGE CYBER setting)	3	All	None	None	CTRL and D (3)	NEXT	LF
IBM 2741	4	All	None	None	None	RETURN	ATTN or RETURN
Teletype Model 40-2 (Conversational mode S/R switch setting only)	5	All	None	None	CNTRL and D (3)	RETURN	NEW LINE (4)
Teletype Model 40-2 (Batch mode S/R switch setting only)	5	None	None	SEND	CNTRL and D (3)	RETURN	NEW LINE (4)
Hazeltine 2000 (HALF or FULL switch setting only)	6	All	None	None	CNTRL and D (3)	CR	LF (5)
Hazeltine 2000 (BATCH switch setting only)	6	None	None	SHIFT and XMIT	CNTRL and D (3)	CR	LF (5)
Digital Equipment VT100 (ANSI X3.64)	7	All	None	None	CNTRL and D (3)	RETURN	LINE FEED

TABLE 2-1. TERMINAL INPUT CONTROL KEYS (Contd)

Terminal	Terminal Class	Function					
		Terminal-to-NPU Transmission			End-of-Block Indicator ([cr])	End-of-Line Indicator ([cr])	End of Physical Line (Line Feed)
		Character (1)	Line (1)	Block (2)			
Tektronix 4010 (teletypewriter emulation)	8	All	None	None	CNTRL and D (3)	RETURN	LF
HASP Workstation	9 or 14	None	Varies (6)	None	None	Varies (6)	None
CDC CYBER 18 COMM 18 HASP Simulator	9	None	RETURN (7)	None	None	RETURN	None
CDC 200 User Terminal (LINE switch setting)	10	None	SEND	SEND	SEND	RETURN or NEW LINE	None
CDC CYBER 18 COMM 18 200 User Terminal Simulator	10	None	RETURN	RETURN	RETURN	RETURN	None
CDC 714-30/40	11	None	ETX	None	ETX	NEW LINE	None
CDC 711-10	12	None	ETX	None	ETX	NEW LINE	None
CDC 714-10/20	13	None	ETX	None	ETX	NEW LINE	None
CDC 731-12 or 732-12	15	None	ETX	None	ETX	NEW LINE	None
CDC 734	15	None	SEND	None	SEND	NEW LINE	None
IBM 2780	16	None	End of card (8)	None	None	EM	None
IBM 3780	17	None	End of card (8)	None	None	EM	None
IBM 3270	18	None	ENTER	None	ENTER	n/a	None

NOTES:

(1) Does not apply if connected through an X.25 packet-switching network.

(2) If connected through an X.25 packet switching network, a key or key sequence defined by the packet assembly/disassembly (PAD) service is always used.

(3) If connected through an X.25 packet-switching network, the end-of-packet sequence M-bit setting also serves this function.

(4) A hardware option can make this key functionally identical to the RETURN key.

(5) This key does not cause a repositioning operation at the terminal.

(6) Different terminals use different keys for this purpose.

(7) Requires COMM 18 command prefix. This emulator uses an exception-processing concept for communication with the network.

(8) Generates end-of-message (EM) code sequence.

A logical line of input always ends when an interactive device transmits an end-of-line indicator or when an interactive device transmits an end-of-block indicator while the input mode is defined as block mode (IN=BK). An upline message is transmitted to the host as soon as a logical line ends.

END-OF-LINE INDICATORS

The end-of-line indicator is initially established by the network administrator when he or she defines the device. The indicator is either a specific code, a code sequence, or a specific condition associated with your use of a certain key or set of keys. The default keys for generating an end-of-line indicator are shown in table 2-1. For mode 4 terminals, the default end-of-line indicator is also the end-of-block indicator, allowing one message per transmission block.

You or the application program can change this indicator (refer to the EL command options, described in section 5). The NPU normally discards any end-of-line indicator character code when it detects the end of a logical line.

MULTIPLE LOGICAL LINES IN ONE MESSAGE

For upline data from a console, the network administrator can configure the device so that the NPU ignores the character or event that normally causes it to transmit a message as soon as a logical line ends. Instead, he or she can make the NPU use a different character or event to trigger transmission to the host. You or the application program can also make this change (refer to the EB option of the EL command, described in section 5).

This option allows you to pack many logical lines into one set of upline network blocks. Each line includes the end-of-line indicator as a data character that terminates it. This is a form of single-message mode, because the host receives only one message. From your viewpoint, one message is many logical lines.

END-OF-BLOCK INDICATORS

The end-of-block indicator is initially chosen for the device by the network administrator when he or she defines the device. The indicator is either a specific code, a code sequence, or a specific condition associated with your use of a certain key or set of keys.

The default keys for generating an end-of-block indicator are shown in table 2-1. In X.25 packet-switching networks, the packet transmission condition is always an end-of-block indicator.

When a console is not operating in block mode, the end-of-block indicator is treated as any other character.

You or the application program can change the end-of-block indicator (refer to the EB command, described in section 5). This indicator normally is discarded when the last message from the device is sent upline.

CONSOLE OUTPUT CONCEPTS

A downline message can contain many logical lines of output. Each logical line can contain many physical lines of output.

A logical line of output ends when the application program embeds a code or set of bytes for that purpose in the message, or when the block containing the line ends. A downline message ends when an application program indicates that condition.

Because downline messages can always contain more than one logical line, a console can always receive the output equivalent of a multiple-message block mode input transmission. The application program can group logical lines as necessary to achieve that effect.

If a message fits into a downline network data block, the block becomes a single-block message. If one downline message cannot be fit into a single network data block, the application program can split it into as many blocks as necessary. An application program generally sends a single message (consisting of as many logical lines as necessary) as the response to one input message from a console.

When you discard a block of output (that process is described later in this section), you might be discarding:

Several physical lines

One logical line

Several logical lines within a longer message

An entire message

BATCH DEVICE DATA

Batch devices can be serviced as site-defined device types through the console interface described in this section. A separate set of information structures also exists for batch devices serviced by CDC-written Terminal Interface Programs and application programs.

These programs require large amounts of data to be exchanged between a host computer's mass storage devices and CDC-defined batch devices. Such batch data is therefore assembled into messages of one or more network data blocks. Each network data block contains one or more mass storage physical record units (PRUs). Because only the CDC-written Remote Batch Facility can use the special interface for CDC-defined batch devices, the remainder of this manual does not discuss the requirements this interface imposes on batch data or batch device support.

CONSOLE DATA

As mentioned earlier in this section, a console can send and receive data in two modes:

Normalized mode

Transparent mode

The format and content of data in these modes are described later in this subsection. The characteristics of a console depend on which data exchange mode is currently used.

In normalized mode, from the application's view, the characteristics of a console are as follows:

A page of output has infinite (no physical) width; logical lines are divided automatically as needed to fit the physical line restrictions of the device.

A page of output has infinite (no physical) length; sets of logical lines are divided automatically as needed to fit the physical restrictions of the device page.

A logical line of output cannot be longer than a single network block; a single message can contain an infinite number of logical lines.

Characters are either 7-bit ASCII codes using zero parity (bit 7, the eighth bit, is always zero in upline data and ignored in downline data), or 6-bit display codes with no parity.

Logical lines of input are terminated by a changeable character or condition; this terminator is the end-of-line or end-of-block indicator described earlier in this section. The input terminator is not part of the data seen by an application program unless the full-ASCII feature is used (this is explained later in this subsection and in section 5 where the FA command is described).

Logical lines of output are terminated by an ASCII unit separator character code (US, represented by the hexadecimal value 1F) or the end of a zero-byte terminated record. The application program places this terminator in the data.

Logical lines can be divided into physical lines by embedding optional format control characters in downline blocks. Output formatting is described in more detail in section 3.

In transparent mode, from the application's view, the characteristics of a console are as follows:

A page of output has infinite (no physical) width.

A page of output has infinite (no physical) length.

Characters are either 7-bit codes using zero parity (bit 7, the eighth bit, is always zero in upline data and ignored in downline data), or codes of a terminal-dependent code set with terminal-dependent parity.

Messages of input are terminated by a changeable character or condition; this terminator is one of the message or mode delimiters described later in this section. The mode delimiter is not part of the data seen by an application program.

Messages of output are terminated by a condition or event chosen by an application program (each network block is separately designated as transparent or normalized when sent).

The application must perform any cursor positioning actions required, and must make any timing adjustments needed at the end of physical output lines.

LINE TURNAROUND CONVENTION

The application program can service a console as if the device always operates in a full-duplex mode. That is, input and output can occur independently; you can enter several logical lines at once (an operation called typeahead), without waiting for a response to each line.

Application program input and output need not alternate. However, some devices cannot actually operate that way. To prevent a loss of synchronization between input and output at such devices, a line turnaround convention exists. This convention consists of the following events.

The end of a message has special significance to the network software during output to a console. When such a block is the last block of output queued for delivery to the device, the network software:

Unlocks the keyboard of a console being serviced as terminal class 4 (an IBM 2741).

Sends an X-ON code to start an automatic paper tape input mechanism, if one has been defined as the input mechanism for the device. Paper tape operation is explained in more detail in sections 3 and 5 where the IN and OP commands are described.

Identifies an automatic input prompt to be returned, if the application program uses this feature. When this feature is used, the network software delivers the block to the device and retains the first 20 characters in the NPU's input buffer. Subsequent input from the device is attached to the end of the retained data. (If more than one logical line is received from the device, the first is appended to the retained data.) All logical lines are transmitted to the host as received from the device.

If the terminal is a half-duplex device, such as a 2741 or a paper tape reader/punch, it must enter input before the network software will deliver additional output messages. Other devices are not subject to this restriction.

You can satisfy the requirement for an input block after an end of message is output in several ways. You can enter an empty input line, which will reach the application program as an end of message for a block containing nothing. You can enter a line containing data, which will reach the application program as one or more network data blocks.

Full duplex devices can interrupt output by entering input. When this occurs, the network software stops the output until you complete the input (using an end-of-line or end-of-block indicator). Output then resumes at the next character of the current physical and logical line.

CONSOLE EXCHANGE MODES

The conventions of block content depend on the mode in which the block is exchanged. There are two possible exchange modes, normalized mode and transparent mode. The latter is referred to in other manuals as binary mode. This manual uses transparent mode to indicate exchange of a block that is not in normalized mode.

Normalized Mode Operation

The network software assembles message character data into upline network data blocks from terminal transmission blocks. It disassembles character data from downline network data blocks, reassembling the characters into terminal transmission blocks. Figure 2-1 illustrates this process.

The assembly operation is controlled by the termination of logical lines. The disassembly operation can be controlled by the termination of physical lines when that is appropriate for the output mechanism of the device. The disassembly operation can also be modified by format control characters embedded in each block, and by the page width defined for the device (refer to the PW command in section 5).

End of Logical Lines in Input

Logical lines reach an application program as one or more network data blocks. Logical lines usually end when a message ends and do not contain the character or code sequence defined as the end-of-line or end-of-block key.

However, two special cases exist. Logical lines do contain the end-of-line or end-of-block codes when the device is operating in full-ASCII editing mode (described later in this section). Logical lines also contain the end-of-line code as data for the host when the network end-of-line indicator is changed to be the default end-of-block indicator for the device (see the EB option of the EL command described in section 5). In the latter case, the transmission block becomes a message, and the logical lines within it have no effect on construction or type of network data blocks.

Logical and Physical Lines in Output

A single logical line can contain many complete physical lines. A single block can contain many complete logical lines, and a message can be one or many such blocks. A physical or logical line cannot, however, be continued from one block to another.

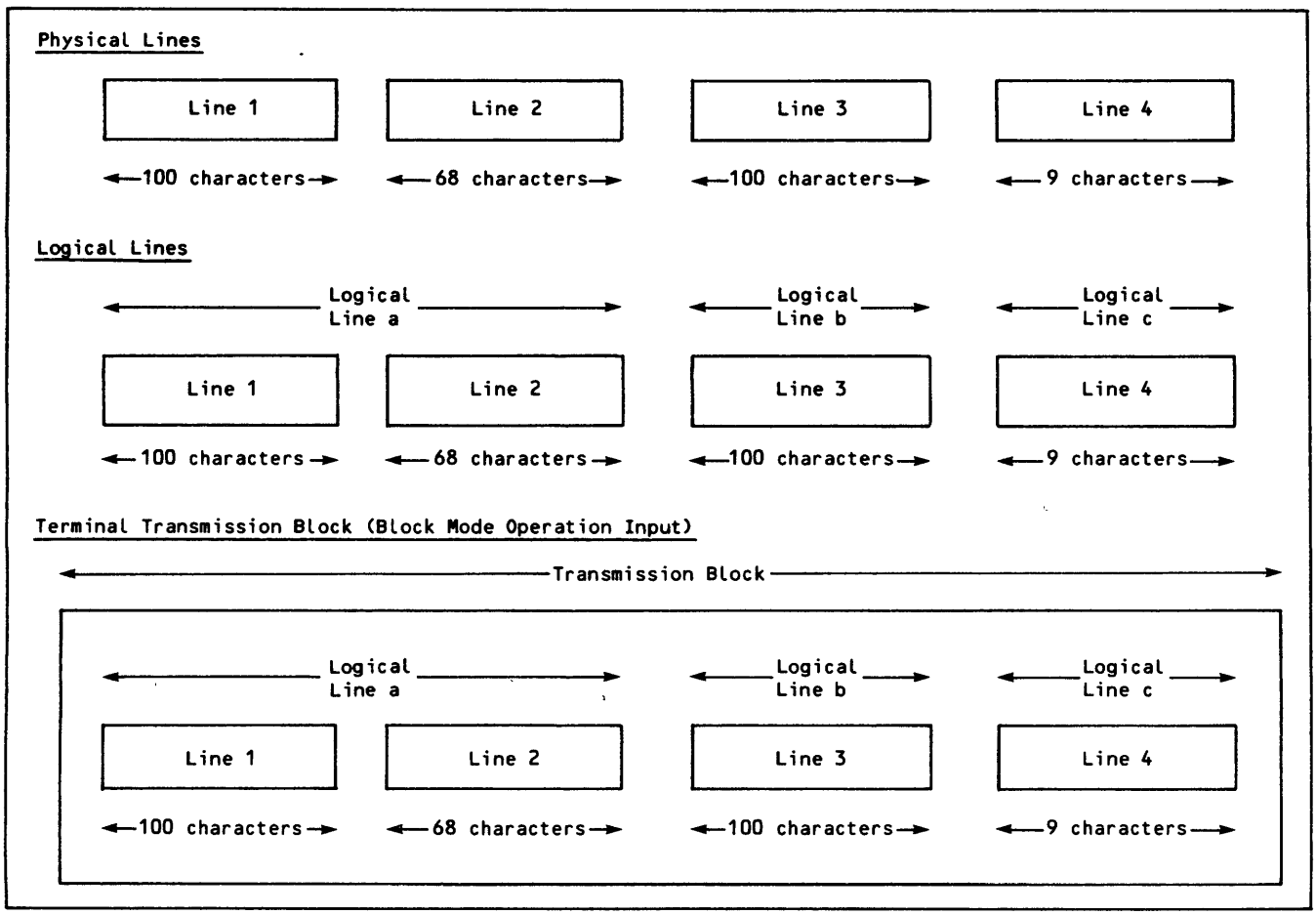


Figure 2-1. Physical and Logical Information Structures in Normalized Data

Logical lines within downline blocks are ended by an end-of-line indicator. Unlike the end-of-line indicators used in upline blocks, downline blocks always contain codes for the end-of-line function; the codes used downline are always the same and usually differ from the codes used upline. The downline end-of-line indicator varies according to the way the application program stores information in the block.

The end-of-line indicator in blocks using 8-bit characters in 8-bit or 12-bit bytes is determined by whether the block is sent in normalized mode or transparent mode (described later in this section). In transparent mode, no end-of-line indicator exists. In normalized mode, the end-of-line indicator is the ASCII unit separator character US.

The end-of-line indicator in blocks using 6-bit character bytes is 12 to 66 bits of zero; these bits are right justified to fill the last central memory word involved. This convention makes each logical line the equivalent of a zero-byte terminated logical record.

The 6-bit option requires a right-justified 12-bit byte in at least one central memory word. On computers using the 64-character set, the colon is represented in 6-bit display code by six zero bits. On such systems, if the application needs to send colons to the console in 6-bit display code, care must be taken to make sure that a string of colons is not interpreted as an end-of-line indicator. A colon preceding the end-of-line indicator is considered as part of the indicator and not as a colon when it occupies one of the two rightmost character positions in the next-to-last central memory word of the block or any of the eight leftmost positions in the last word of the block.

All predefined end-of-line indicators embedded within a block are discarded by the network software and produce no characters on the console. The network software can perform carriage or cursor repositioning when an end-of-line indicator is encountered; this operation is in section 3 under Output Formatting, where format effectors are described.

Upline Character Sets and Editing Modes

The network protocol permits entry from a device of any code less than or equal to 8 bits per character; however, a normalized mode character always reaches an application program as one of the 128 ASCII characters defined in appendix A. Receipt of an entered character by the application program depends on the editing functions performed by the TIP. Three editing modes exist for the TIP when it processes normalized data:

Complete normalized editing mode

Special editing mode

Full-ASCII mode

Devices usually begin a connection with the network in normalized mode. The initial upline editing mode is complete editing. You or the application program can change that mode using the SE or FA commands, described in section 5.

Complete Editing

During complete editing operations, the following hexadecimal character codes cannot be received by the network application program:

00 (the ASCII character NUL)

0A (the ASCII character LF)

7F (the ASCII character DEL)

The backspace character code currently defined for the device (see the BS command in section 5)

The end-of-line indicator currently defined for the device (see the EL command in section 5)

The end-of-block indicator currently defined for the device (see the EB command in section 5)

The following hexadecimal character codes cannot be received, if entered at certain points in a message:

02 (the ASCII character STX), if entered as the first character of a message

11 (the ASCII character DC1) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if the input mechanism is known to be a paper tape reader (see the PT option of the IN command in section 5)

The user-break-1 and user-break-2 character codes currently defined for the terminal, if entered as the only character in a message (see the B1 and B2 commands in section 5)

The abort-output-block character code currently defined for the terminal, if entered as the only character in a message (see the AB command in section 5)

The network control character currently defined for the terminal when it follows an end-of-line or end-of-block indicator or when it is used for such purposes as page turning (see the CT command and the Y option of the PG command in section 5)

The currently defined cancel-input character is always received at the end of the logical line it cancels. This character is not data.

Special Editing

Special editing takes precedence over complete editing. Special editing cannot occur if the terminal operates in network block mode.

When special editing occurs, line feed codes and the currently defined backspace code are forwarded to the application program as data. The network software sends appropriate responses to the device when it receives these codes.

During special editing operations, the following hexadecimal character codes cannot be received by the network application program:

00 (the ASCII character NUL)

7F (the ASCII character DEL)

The end-of-line indicator currently defined for the device (see the EL command in section 5)

The end-of-block indicator currently defined for the device (see the EB command in section 5)

The following hexadecimal character codes cannot be received, if entered at certain points in a message:

02 (the ASCII character STX) if entered as the first character of a message

11 (the ASCII character DC1) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if it follows an end-of-line or end-of-block indicator and the input mechanism is known to be a paper tape reader (see the PT option of the IN command in section 5)

The user-break-1 and user-break-2 character codes currently defined for the terminal, if entered as the only character in a message (see the B1 and B2 commands in section 5)

The abort-output-block character code currently defined for the terminal, if entered as the only character in a message (see the AB command in section 5)

The network control character currently defined for the terminal when it follows an end-of-line or end-of-block indicator or when it is used for such purposes as page turning (see the CT command and the Y option of the PG command in section 5)

The currently defined cancel-input character is always received at the end of the logical line it cancels. This character is not data.

Full-ASCII Editing

Full-ASCII editing takes precedence over special editing or complete editing. When full-ASCII editing occurs, almost all codes are forwarded to the application program as data. The network software does not perform actions at the terminal when it receives the codes for backspace, abort-output-block, cancel-input, user-break-1, or user-break-2. These codes and the end-of-line and end-of-block indicator codes are sent upline as data.

During full-ASCII editing operations, the following hexadecimal character codes cannot be received by the network application program:

00 (the ASCII character NUL) if it occurs after the end-of-line or end-of-block indicator

0A (the ASCII character LF) if it occurs after the end-of-line or end-of-block indicator

7F (the ASCII character DEL) if it occurs after the end-of-line or end-of-block indicator

11 (the ASCII character DC1) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if the TIP is supporting output control for the device (see the Y option of the OC command in section 5)

13 (the ASCII character DC3) if it follows an end-of-line or end-of-block indicator and is explicitly supporting paper tape input from the device (see the PT option of the IN command in section 5).

The network control character currently defined for the terminal when it follows an end-of-line or end-of-block indicator or when it is used for such purposes as page turning (see the CT command and the Y option of the PG command in section 5)

The currently defined cancel-input character is always received as the last character of the logical line it ended. This character is data.

Downline Character Sets

The network protocol permits transmission from a network application program of any character code less than or equal to 8 bits. If the application

program transmits an 8-bit code, it cannot use the upper (eighth) bit for data unless it is transmitting in transparent mode.

In normalized mode, the application program can only use the 128 ASCII characters defined in appendix A. If the application program transmits a 7-bit ASCII code, it cannot use the upper (eighth) bit for parity; the network ignores the eighth bit in downline normalized mode data.

Receipt of a transmitted character by the device depends on the editing functions and character transformations performed by the TIP. In addition to character codes altered during the translation and substitution operations described elsewhere in this section and in appendix A, the hexadecimal character code 1F (the ASCII character US used as a downline block end-of-line indicator) cannot be received by a device when the application program transmits a block in normalized mode.

If the application program allows format effectors in output and does not supply them automatically, the first character in each logical line also is deleted or replaced. Format effectors are described in section 3.

Page Width and Page Length

The application program receives an indication of the page width and page length in effect for a console when connection with the device first occurs. You or the application program can change the page width and page length in effect for the console.

The Terminal Interface Program uses the page width value to transform logical lines of downline data into physical lines of output and to determine the boundaries allowed for backspacing. The TIP uses the page length defined for the console to format physical lines into physical pages or screens of output.

For consoles defined as having hardcopy output mechanisms (see the PR option of the OP command in section 5), a logical line of downline data containing more characters than the page width value permits is divided into singly spaced physical lines. These physical lines are equal to or shorter than the page width in effect and are displayed successively.

For all consoles, the page width is used as part of the line-counting algorithm to determine the page length. Each logical line is examined to determine how many multiples of the page width (how many physical lines) it contains. Each complete or partial multiple counts as one line when the TIP determines the page length.

Line counting begins at the beginning of each downline message. The line counter is reset to zero each time the page length of the device is reached, each time any input occurs, or when page turning occurs during page waiting operation. Refer to the PG, PW, and PL commands in section 5.

The physical line width of the device might be smaller than the page width defined for the device. When this happens, the effect of sending a logical line of downline data containing more characters than the physical line width permits depends on the terminal hardware.

Transparent Mode Operation

Blocks exchanged between an application program and a console device in transparent mode do not use most of the features of the network:

- No input editing occurs except, by option, the Asynchronous, Mode 4, and X.25 TIPs can recognize the user-break-1 and user-break-2 character codes currently defined for the terminal if entered as the only character in a message (see the B1 and B2 commands described in section 5).

- No code conversion occurs.

- No format effector deletions and replacements are performed for downline blocks.

- No page width operations are performed to preserve physical line boundaries.

- Page waiting occurs only at the end of a downline message (see the PG command in section 5).

Transparent mode operation is separately selected for input and output. Either you or the application program can start transparent mode input, using the IN command described in section 5. Only the application program can start transparent mode output.

Transparent mode messages need not be equivalent to normalized mode logical lines.

Transparent mode data can occupy up to 8 bits of an 8-bit byte, representing up to 256 distinct character codes or device instructions. Codes longer than 8 bits cannot be exchanged; data packed in 12-bit bytes by an application program or a terminal device is truncated to 8 bits by the network software.

HASP terminals (terminal classes 9 and 14) and bisynchronous terminals (terminal classes 16 and 17) cannot transmit or receive such blocks. All other terminals can, although mode 4 terminals (terminal classes 10 through 13 and 15) require the special treatment described below.

During transparent mode operation, the application program is responsible for all data formatting and terminal control. For mode 4 terminals, this means that the Terminal Interface Program does not blank-fill the current line and unlock the keyboard before input can be performed; the TIP does add or remove the line transmission portion of the protocol envelope (characters that are not part of your data) to or from all message text exchanged with the terminal.

Two mutually exclusive forms of transparent mode input can be selected. The network administrator can make this selection when the device is defined in the network configuration file, or you or the application program can make it while the device is active. The two forms are:

Single message

Multiple message (analogous to block mode operation)

Single-Message Input

For single-message input, one or more transparent mode input delimiters are specified, using the DL command options described in section 5. The end of a message ends transparent mode input. The opposite is true also: when transparent input ends, so does the current message. Single-message transparent mode input ends when the Terminal Interface Program encounters one of the mode delimiter conditions.

Multiple-Message Input

For multiple-message input, you or the application program define:

One or two input end-of-message indicators (equivalent to a normalized mode end-of-line indicator)

One or two transparent mode input delimiters (analogous to a normalized mode end-of-block indicator)

Each message ends at an end-of-message indicator; the last message ends when transparent input mode ends. The end of a message, however, does not automatically mean the end of transparent mode input. The end-of-message indicator and mode delimiters are specified using the XL command options in section 5.

Upline Message Blocks

A transparent mode input block is assembled each time the network block size is reached or the Terminal Interface Program encounters an end-of-message indicator. The last block in the last message is assembled when the delimiter condition is encountered. Figure 2-2 illustrates this concept.

If the end-of-message indicator is a specific character code, the TIP removes that code from the character stream before assembling the last block. The transparent mode delimiter code (if one exists) is also removed.

In transparent mode, the concept of a logical line is not meaningful to the network software. Both the end-of-line and end-of-block indicators are data within a transparent message.

Transparent Mode Output

Transparent mode output data can be divided arbitrarily into blocks and messages, provided the restrictions on network block size are met. A transparent mode downline block ends when the last character it contains is transferred to the network.

If the TIP is performing page-wait operations for the terminal during transparent mode operation, output stops to wait for terminal operator acknowledgment at the end of each message. The automatic input feature can be used with the last block of a transparent mode output message.

Parity Processing

Actual terminal codes are right-justified with zero fill within the 8-bit character portion of the input or output byte. The codes contained in the input or output bytes depend on the parity option declared for the terminal.

The actual terminal code parity bit can be used for meaningful code only if no parity, or ignore parity, is declared (see the N and I options of the PA command in section 5). Otherwise, the parity bit is zero in input blocks and set by the network processing unit on output.

For example:

If the terminal uses a 7-bit code such as ASCII, with the eighth bit as a parity bit, the setting of the eighth bit is determined by the parity option selected for the terminal. If zero parity is declared, the eighth bit is always zero on input and output. If odd or even parity is declared, the eighth bit varies on input and output to satisfy the character parity requirement. If no parity, or ignore parity, is declared, the eighth bit is treated as part of the character data and is not changed during input or output.

If the terminal uses a 6-bit code, with the seventh bit as a parity bit, the setting of the seventh bit is determined by the parity option selected for the terminal. If zero parity is declared, the seventh bit is always zero on input and output. If odd or even parity is declared, the seventh bit varies on input and output to satisfy the character parity requirement. If no parity, or ignore parity, is declared, the seventh bit is treated as part of the character data and is not changed during input or output.

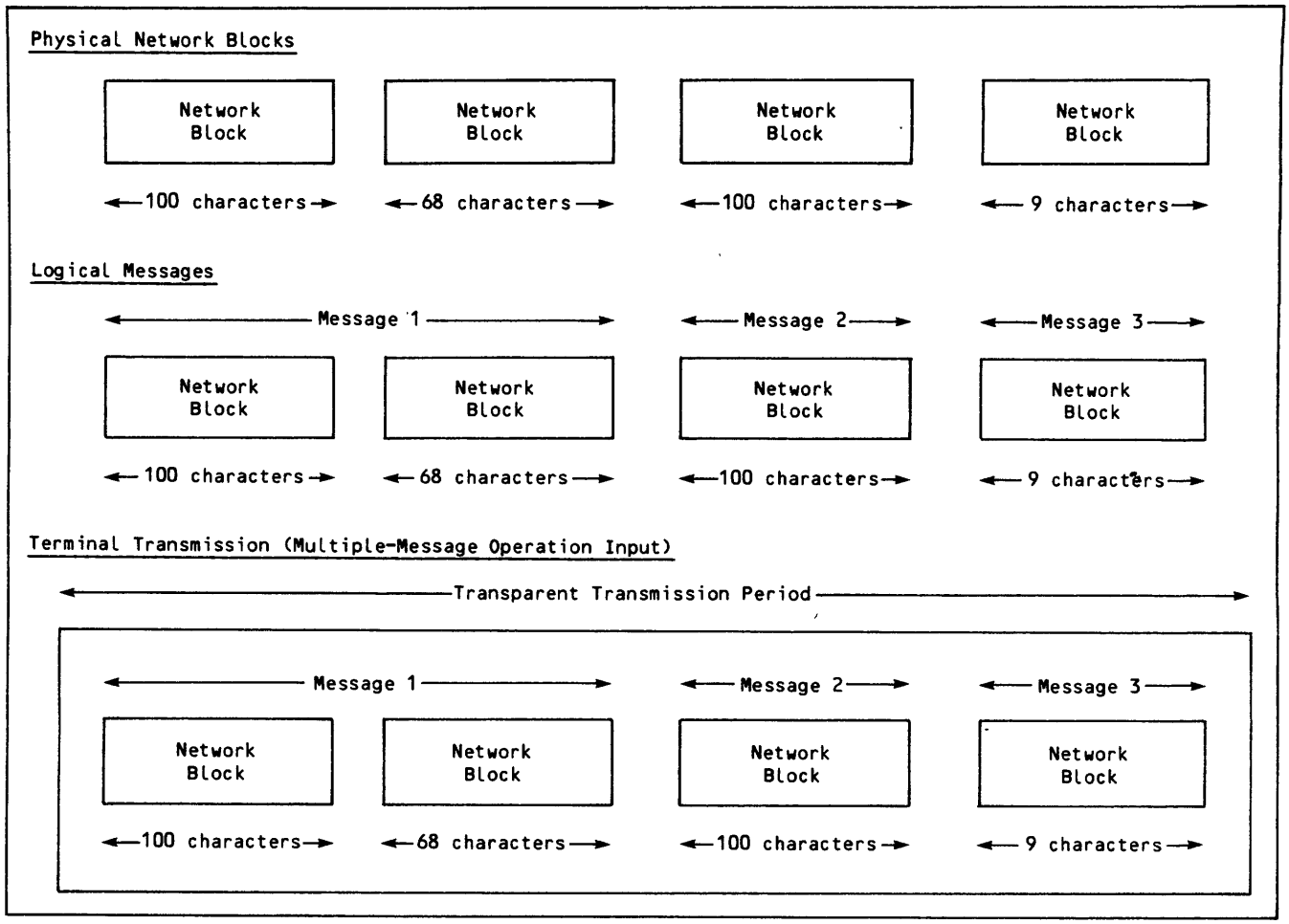


Figure 2-2. Physical and Logical Information Structures in Transparent Mode Data

USING THE INTERFACES

We can now examine how you use this information when:

Transmitting data from a terminal

Receiving data at a terminal

DATA INPUT TRANSMISSIONS

Bewildering things sometimes occur when a terminal user presses a key. Pressing a key at one terminal causes action A, while pressing a key marked the same way at another terminal causes action B (or, worse, yet, actions B, C, and D).

You can understand these situations, and sometimes avoid them, by knowing the following:

When a character is transmitted by the terminal

When a line is transmitted to the host computer

What constitutes a physical line of data

What constitutes a logical line of data

What constitutes a logical message

The network's view of these things was discussed earlier in this section. Now we can examine how you use these features through a console keyboard.

Transmission Time and Transmission Block Size

When transmission of data from the terminal occurs depends on the upline transmission block size of the terminal. On some terminals, transmission time and upline transmission block size are selected with a hardware or controlware switch (usually described in the terminal operator's guide as a transmission mode switch). At other terminals, the transmission time and size are fixed when the terminal is built.

There are three transmission times, with corresponding transmission block sizes:

Character-by-character (called conversational mode or character mode)

Line-by-line (called line mode)

Block-by-block (called batch mode or block mode)

When your terminal uses character mode transmission, each character is transmitted to the network processing unit as soon as you enter it. A transmission block is one character long.

When your terminal uses line mode transmission, each character you enter is stored in the terminal buffer memory until you indicate that the line is complete (or until the buffer is completely filled). Transmission to the network processing unit occurs only when no more information can be entered on the line. A transmission block is typically 80 characters long.

When your terminal uses block mode transmission, each character you enter is stored in the terminal buffer memory. When you indicate that the line is complete, that indication is also stored. Unless you also indicate that the buffer can be transmitted to the network processing unit, any characters you subsequently enter are stored within the terminal as part of another line of data. A transmission block can be several thousand characters long.

In summary, character mode terminals do not buffer input before transmission. Line mode terminals buffer an entry one physical or logical line long before transmission. Block mode terminals buffer many entries of physical or logical lines before transmission.

Character mode terminals for which paper tape input has been prepared off-line operate like network block mode terminals. Teletypewriter use of paper tape in this manner is described in section 3.

The primary impact of transmission time, from a terminal user's viewpoint, is on editing of input. Mistakes can be edited out of entries on block mode or line mode terminals simply by repositioning the cursor or typing element carrier (usually by backspacing) and typing over the error. The erroneous character, backspace characters, and characters backspaced over are all discarded before transmission. This procedure does not work on entries at character mode terminals, because the information has already left the terminal.

One function of the network processing unit is to make input from one terminal look the same to host software as input from any other terminal. The NPU accomplishes this by buffering characters received from all terminals. This buffering permits the NPU to edit normalized mode input in a manner that mimics the off-line editing performed at line mode or block mode terminals.

The NPU transmits the buffered input to the host only when you enter a code (such as the end-of-line or end-of-block indicator) that says no more data will be input for that line. This indicator terminates buffer filling and sends a data message to the host.

The network processing unit can distinguish between ending a physical line of data and ending a message. It makes this distinction on the basis of the character entered at the end of the line.

If you end a physical line of data by pressing a line feed key, the network processing unit does not usually send the accumulated characters to the host, but presumes that the characters are part of a longer message. The NPU responds by causing a carriage return operation at the terminal immediately after the line feed caused by the key.

The NPU does not process line feed characters this way when you use the special editing feature or change the blocking factor. Special editing was described elsewhere in this section and in section 5; the blocking factor is discussed with the BF command in section 5.

If you end the line of data by entering an end-of-line or end-of-block indicator, the NPU sends the accumulated characters to the host and indicates that the message is complete. The NPU can inform you of this operation by causing a line feed operation and/or a carriage return operation in response to the indicator received.

These conventions give you four distinct function keys affecting transmissions from you to the host:

- A physical line forwarding (network data block transmission) key recognized by the network
- A message termination and transmission (end-of-line or transparent mode end-of-message) key recognized by the network
- A block termination and message transmission (end-of-block or transparent mode delimiter) key recognized by the network
- A message or block transmission key recognized by your terminal

At character mode and line mode terminals, the second and third functional keys are usually the same physical key; terminating a message transmits that message.

At a block mode terminal, the message termination key can be different physically as well as functionally from the message transmission key. You must understand the distinction between these keys to know when the NPU and the host software perform their functions for messages from such terminals. The NPU cannot transmit a terminated message to the host until the terminal transmits the message to the NPU.

At line mode and block mode terminals, there is another factor determining when transmission from the terminal occurs. When such terminals use controlware or an emulator, the controlware or emulator can treat console entries in two ways:

- As communication intended for the host computer, unless otherwise indicated; this mode of operation is called direct-transmission processing in this manual.
- As communication intended for the emulator, unless otherwise indicated; this mode of operation is called exception processing in this manual.

Direct-transmission terminals are the most common among the terminals serviced by the network. Such terminals always transmit information to the network when the transmission key is pressed.

Exception-processor terminals do not transmit anything to the network when the transmission key is pressed, unless the entry has been prefixed by something that serves as a command to make such a

transmission. Such terminals transmit information to the network only when the form of each console entry is:

prefix information [cr]

All entries described in this manual assume that you are operating a direct-transmission terminal. Operators of exception-processor terminals must adapt the format of an entry description to match the prefixing requirements of their specific terminals.

Because of the ways in which your terminal can transmit data and the network can divide it into network data blocks, an application program can receive three types of input transmissions:

A single-block message

One block of a multiple-block message

Multiple messages of one or more blocks each

Not all application programs can distinguish among these three types. Because the single-block message is the simplest type and must be recognized by any application program to be functional, most manuals describing terminal use specify ending an entry by pressing the terminal transmission key. This manual is no exception.

You should not use a message termination key that is separate from a message transmission key unless the application program receiving the transmissions supports the concept known as typeahead. Typeahead always exists when multiple messages are transmitted in a single terminal transmission block from the terminal. You should not use a line feed key as a line forwarding key unless the application program receiving the transmissions accumulates all transmissions constituting a single message before processing the first transmission of the set.

There are two occasions when the distinction between transmitting a line and transmitting a message is important to you:

When transmitting a line suspends input processing, but transmitting a message does not

When transmitting a line suspends output processing, but transmitting a message does not

If you are communicating with a network application program that is designed to process messages, transmitting a line rather than a message causes the application program to delay processing of the entry until the message is complete. This delay can cause a situation in which you are waiting for the application program to do something, while the application program is waiting for you to do something.

The network processing unit always gives input precedence over output. If input begins while a line of output is in progress, the network processing unit saves the output line until input of the message is complete, then resumes output from the point of interruption.

If input begins while output is available for the terminal but output delivery has not begun, the output is saved. This means that transmitting an input line indefinitely delays all output to the terminal until the terminal transmits the last line of the message. Any response by an application program to a transmitted line is therefore delayed until a complete message is transmitted.

The use of terminal line feed, message termination, and message transmission keys is summarized in the following subsection. The key identified in this subsection with a specific function does not always match the key on the terminal being used, because various terminals can operate in the same terminal class.

Line Feed and Transmission Keys

Table 2-1 shows the default keys used by the supported terminal classes to perform the end-of-line, end-of-block, line feed, and terminal transmission functions. The keys shown are for typical terminals in each terminal class. The terminal classes are described in section 5.

Some manuals divide the keys listed in table 2-1 functionally into three types and refer to them as carriage return, new line, and line feed. These keys are described in two ways: functions they provide at the terminal, and interpretations an application program makes when they are used.

Terminal functions are:

- Carriage return

  This key: terminates a message if it is defined that way; advances the cursor or carriage to the beginning of the next line; and transmits the message if it is defined that way. It also transmits any other messages that may have been stored in the terminal buffer. This key generates an end-of-line or an end-of-block indicator in normalized mode and is the one normally referred to by the term [cr] in this guide. A key generating a new line (NL) ASCII code can be used if available.

- New line

  This key: terminates a message if it is defined that way, but does not transmit the message. Rather, it causes the message to be stored in the terminal's buffer until the carriage return key is used to transmit the entire buffer to the host. The new line key is an end-of-line key when used at a block mode terminal.

- Line feed

  This key: can transmit one physical line from the NPU buffer as a network block; advances the cursor or carriage to the beginning of the next line; does not terminate the message. Subsequently using the carriage return or new line key causes the message to be terminated.

The application program interprets use of these keys as:

- Carriage return

  This key terminates a message.

- New line

  This key terminates a message. Correct processing of subsequent messages in the same transmission depends on whether the application program supports typeahead dialog.

- Line feed

  This key does not generate data unless partial suppression of input editing (special editing) has been selected. Otherwise, the key can terminate a single transmission. If the application program distinguishes between transmission of a line and transmission of a message, this key leaves the program expecting additional input.

  When an application program permits entry of parameters for a single operation as successive transmissions, the line feed key can function as a delimiter.

Input Editing Keys

Because of the buffered nature of some terminals, there are two times when you can edit the contents of a message:

Before transmission from the terminal

Before transmission from the network processing unit

Character mode terminals do not permit editing before transmission from the terminal. Line mode and block mode terminals do. You have several choices of keys for editing from line mode or block mode terminals. Table 2-2 summarizes these input editing keys for terminals in all terminal classes.

TABLE 2-2. INPUT EDITING KEYS

Terminal	Terminal Class	Editing Within Terminal			Editing Within Network		
		Delete Character	Delete Line	Delete Message	Delete Character	Delete Line (1)	Delete Message (1)
Teletype Model 30 Series	1	None	None	None	(H) (2)	(X) (2)	(X) (2)
CDC 713	2	None	None	None	<-- (2)	(X) (2)	(X) (2)
CDC 751 or 752 or 756 (CHARacter switch setting only	2	None	None	None	<-- (2)	(X) (2)	(X) (2)
CDC 751 (LINE switch setting only)	2	<--	LINE CLEAR	LINE CLEAR	(<--) (2)	(X) (2)	(X) (2)
CDC 751 or 756 (BLOCK switch setting only)	2	<--	LINE CLEAR	CLEAR	(<--) (2)	(X) (2)	(X) (2)
CDC 721 Viking Extended	3	None	None	None	<-- (2)	(X) (2)	(X) (2)
IBM 2741	4	None	None	None	BACK SPACE (2)	ATTN ((2)	ATTN ((2)
Teletype Model 40-2	5	<-- or CHAR DLETE	LINE DLETE	n LINE DLETE	None	(X) (2)	(X) (2)
Hazeltine 2000 (HALF or FULL switch setting only)	6	<--	None	None	(H) (2)	(X) (2)	(X) (2)
Hazeltine 2000 (BATCH switch setting only)	6	<-- or D/C	D/L	CLR/FG	(H) (2)	(X) (2)	(X) (2)
Digital Equipment Corporation VT100 (ANSI X3.64)	8	None	None	None	BACK SPACE (2)	(X) (2)	(X) (2)

TABLE 2-2. INPUT EDITING KEYS (Contd)

Terminal	Terminal Class	Editing Within Terminal			Editing Within Network		
		Delete Character	Delete Line	Delete Message	Delete Character	Delete Line (1)	Delete Message (1)
Tektronix 4014	8	BACKSPACE or (H)	None	None	(H) (2)	(X) (2)	(X) (2)
HASP Workstation	9 or 14	Varies (3)	Varies (3)	Varies (3)	None	((2)	((2)
CDC CYBER 18 COMM 18 HASP Simulator	9	<--	RUBOUT	CLEAR	None	None	((2)
CDC 200 User Terminal	10	BKSP	None	CLEAR	None	None	((2)
CDC CYBER 18 COMM 18 200 User Terminal Simulator	10	<--	RUBOUT	CLEAR	None	None	((2)
CDC 714-30/40	11	<--	LINE CLEAR	CLEAR	None	None	((2)
CDC 711-10	12	<--	LINE CLEAR	CLEAR	None	None	((2)
CDC 714-10/20	13	<--	LINE CLEAR	CLEAR	None	None	((2)
CDC 731-12	15	<--	LINE CLEAR	CLEAR	None	None	((2)
CDC 732-12	15	<--	LINE CLEAR	CLEAR	None	None	((2)
CDC 734	15	<--	LINE CLEAR	SELECT CLEAR	None	None	((2)
IBM 2780	16	None	None	None	None	None	None
IBM 3780	17	None	None	None	None	None	None
IBM 3270	18	<--	ERASE EOF	None	None	None	((2)

(1) The effect of entering this character depends on the network application program being used.

(2) This is the character most frequently configured for terminal use; this character can be changed by the site administrator or by the terminal operator through a command described in section 5.

(3) Different HASP terminals use different keys for this purpose.

As the table indicates, there are three forms of input editing possible:

Removal of a character from the input

Removal of a line from the input

Removal of a message from the input

When the editing is performed within the terminal *before transmission*, the character identifying the key for the function is fixed. When the editing is performed within the network software after transmission from the terminal, the character identifying the key for the editing function can be changed. The method used to make the change (the BS or CN command) is described in section 5.

Removing a character from the input is usually synonymous with backspacing, and is therefore performed by the NPU when a backspace character is entered. Removing a logical line from the input is synonymous with canceling a message, and is therefore performed by the host when a cancel-input character is entered. Message-removal editing might not be successful if the host application program does not recognize the function.

The following subsections describe what the NPU does when you enter one of these characters. Use of the special-editing and full-ASCII options, which partially suppress input editing by the NPU, are also discussed. Selection of these options is also described in section 5.

Backspacing

Normal editing by the NPU deletes an input character from the NPU buffer each time a character code associated with the backspace function is received. When the default backspacing character is used, pressing the backspace key moves the cursor or carriage back one character position. You can then overtype a new character to replace the existing character in the input.

When your input line is longer than the page width defined for your console, you cannot backspace across the page width boundary. For example, if your console has a page width of 60 characters and you enter 72 characters of data, you cannot backspace more than 12 characters.

NPU processing of backspace characters prevents the following:

Transmitting a backspace character code to the application program in the host computer as data.

Overstriking one character with another to create a third character.

Seeing the component characters when you perform overtyping (the second character either replaces the first or renders it unreadable).

If you need to do any of these things, you can either change the backspace character used by the network or use the special editing option. The NPU supports an alternate form of backspace processing as part of its special editing option. Selection of the option partially suppresses the editing of input.

When you use the special editing option, the NPU reacts differently to receipt of a character code associated with the backspace function. From your viewpoint, pressing the backspace key while special editing is in effect stores the character code associated with the backspace function as data to be transmitted to the application program in the host computer.

If you press the line feed key immediately after pressing the backspace key, and if the terminal is in a class that permits output of a bell character code, the NPU sends such a code. Whether you hear a bell or buzzer noise as a result of the code depends on the hardware options installed at your terminal.

When you use normal input editing, backspaces delete characters. Deletion of anything when special editing is used becomes the function of the application program you are communicating with. The following examples clarify this distinction.

Suppose you press the keys A, Q, <--, P, and L, in that order. If you are using normal input editing, you see something like:

```
    APL
or
    AQL
```

and the application program sees only the three characters APL. If you are using the special editing feature, pressing the same keys makes you see the same thing; however, the application program sees all five characters. If the application program can do so, it will interpret the sequence of Q, backspace, P as either P alone, or a new hybrid character created by an overstrike.

Suppose you press the keys A, Q, <--, LINE FEED, P, and L, in that order. If you are using normal input editing, you see something like:

```
    AQ
    PL
```

and the application program sees one or two transmissions containing the characters APL. However, if you are using the special editing feature, pressing the same keys makes you see something like:

```
    AQ
     PL
```

and the application program sees one transmission containing codes for all six keys.

Unless otherwise indicated, this manual assumes that special editing is not being used. All discussions of either a backspace or line feed operation depend on this assumption.

Canceling a Logical Line

Editing by the host should delete a logical line from unprocessed input each time a character code associated with the cancel-input function is received as the last character of an entry. From your viewpoint, pressing one key (usually the (X) or left parenthesis key) and pressing the end-of-message or end-of-block key allows you to start the line over without backspacing to the beginning of the line. Your new entry line begins at the first position of the line following a line reading:

DEL

that is output by the NPU.

The method of processing the (, (X), or comparable character:

Transmits the cancel-input character code to the application program in the host computer as the last data character of an entry. This allows the program to see what you deleted.

Allows the application program to reissue prompting information, such as auto-input characters, that made up part of the canceled line. Auto-input characters are described in more detail in the next subsection.

DATA OUTPUT TRANSMISSIONS

Data output to a terminal parallels data input in several ways. A network application program can transmit output data in either single line messages or multiple line messages. The network processing unit (at the application program's request) can edit character mode output data to insert terminal character code sequences that reformat lines of output. You can use certain terminal keys to exercise some control over the output transmission process.

Blocks, Lines, and Messages

The distinction among a block of output, a line of output, and a message of output is significant to you in the following contexts:

During use of the abort-output-block feature

During choice of the method used to enter data from the terminal

During paging operations

When a message of output begins, the following occurs:

Terminals with keyboards that can be locked have the keyboard locked to prevent unintentional input that destroys output.

The line counter used by the network processing unit during paging operations begins incrementing from zero.

When a message of output ends, the following occurs:

Locked keyboards are unlocked because input is expected.

Until an input message is received for transmission to the host, no additional output from the host can be delivered to some terminals. Once an output message ends, another usually cannot be delivered until input occurs.

Terminals explicitly using paper tape equipment are sent a code that turns the tape reader on.

When a line of output begins, the network processing unit usually causes an appropriate formatting action on the output device of the terminal. This normally consists of returning the cursor or output mechanism to the left margin of the next line if it is not already positioned there.

When the output line is sent to a batch printer device (serviced separately from the terminal console), the first character of each line can be transformed by the network processing unit as a carriage control code, and discarded from the output data. The carriage control effects of these characters when sent to a printer are shown in tables in section 3. Untransformed data is always printed.

When a normalized mode output line is sent to a console, the first character in each line can be similarly processed, if the application program indicates this is required. Processing of these format effector characters consists of deleting the character from the output line and causing the corresponding operation described in tables in section 3. If the first character of an output line disappears or line folding apparently occurs one character before it should, the application program has erroneously indicated that its message lines contain format effector characters. Whether you need to (or can) use this format effector feature depends on the application program with which you are communicating. When the application does not use the format effector feature and you cannot use it, all output is single-spaced between lines. Additional vertical spacing in such cases occurs when the output message contains blank or empty lines.

Interrupting and Discarding Output

While output is in progress, you can interrupt it at some terminals so that input can be performed. Such terminals possess keys that do not generate character codes when pressed. These keys are used to interrupt the output flow to the terminal, and input is entered immediately after pressing them. These keys are shown in table 2-3.

TABLE 2-3. TERMINAL OUTPUT FUNCTION KEYS

Terminal	Terminal Class	Interactive Output Interrupt Key	Batch Output Interrupt Key	Terminal Buffer Clearance Key	NPU Buffer Clearance Key (1)
Teletype Model 30 Series	1	BREAK	None	None	(P) or (T)
CDC 713	2	BREAK	None	None	(P) or (T)
CDC 751 or 752 or 756 (CHAR switch setting only)	2	BREAK	None	None	(P) or (T)
CDC 751 (LINE switch setting only)	2	BREAK	None	LINE CLEAR	(P) or (T)
CDC 751 or 756 (BLOCK switch setting only)	2	BREAK	None	CLEAR	(P) or (T)
CDC 721 Viking Extended	3	BREAK	None	CLEAR ⇧ (P)	(P) or (T)
IBM 2741	4	ATTN	None	None	: or)
Teletype Model 40-2 (Conversational mode S/R switch setting only)	5	INTRPT	None	None	(F) or (T)
Teletype Model 40-2 (Batch mode S/R switch setting only)	5	INTRPT	None	CLEAR	(F) or (T)
Hazeltine 2000 (HALF or FULL switch setting only)	6	BREAK	None	None	(P) or (T)
Hazeltine 2000 (BATCH switch setting only)	6	BREAK	None	SHIFT and CLR	(P) or (T)
Digital Equipment VT100 (ANSI X3.64)	7	BREAK	None	ESC [P 2 J	(P) or (T)
Tektronix 4014	8	BREAK	None	None	(P) or (T)
HASP Workstation	9 or 14	Varies (2)	None	Varies (2)	: or)
CDC CYBER 18 COMM 18 HASP Simulator	9	(G)	None	CLEAR	: or)
CDC 200 User Terminal	10	None	INTER (hold down until effective)	CLEAR	: or)
CDC CYBER 18 COMM 18 200 User Terminal Simulator	10	(G)	(G)	CLEAR	: or)
CDC 714-30/40	11	None	None	CLEAR	: or)
CDC 711-10	12	None	None	CLEAR	: or)
CDC 714-10/20	13	None	None	CLEAR	: or)
CDC 731-12	15	None	F0	CLEAR	: or)
CDC 732-12	15	None	F0 (hold down until effective)	CLEAR	: or)

TABLE 2-3. TERMINAL OUTPUT FUNCTION KEYS (Contd)

Terminal	Terminal Class	Interactive Output Interrupt Key	Batch Output Interrupt Key	Terminal Buffer Clearance Key	NPU Buffer Clearance Key (1)
CDC 734	15	None	SEND key with firmware controller key in ENABLE position	CLEAR	: or)
IBM 2780	16	None (card reader stops)	None	None	None
IBM 3780	17	None (line printer stops)	None (line printer or card reader stops)	None	None
IBM 3270	18	None	None	None	Program attention 1, Program attention 2, : , or)

(1) These are the characters most frequently configured for terminal use; these characters can be changed by the site administrator or by the terminal operator through a command described in section 5. Entry of these characters during output also causes a user-break-1 or user-break-2 indication to be sent to the application program.

(2) Different terminals operating under HASP protocol use different keys for this purpose.

You should be cautious when using an interactive output interrupt key. If you hold the key down too long at some terminals, you can disconnect the terminal from the network. Your terminal can also be configured so that the interactive output interrupt key is a cancel-input or user-break-1 key; refer to the BR command description in section 5.

If your terminal does not have an interactive output interrupt key, you can use page waiting as a substitute. The pauses at page boundaries allow you to perform any action that requires interruption of interactive output. The network software page wait feature is described in section 5.

When interactive output is interrupted, any partially transmitted line is saved. When the input message is completed, output is resumed. The remainder of the saved line is transmitted.

You can discard one or more lines of interactive output by interrupting output of that line and entering a character identified as the abort-output-block character for the console. This character is described more completely in section 5. Entering the abort-output-block character for the terminal discards only the current block of the message; the remaining lines of the message, if any, are output after the block is discarded.

If correct dialog with the application program after the discarding operation requires the presence of these auto-input characters, you must reenter them.

You can discard an entire output message by interrupting output of that message and entering a character identified as the user-break-1 or user-break-2 character for the console. These characters are described more completely in section 5.

Entering either of these characters discards the entire message being transmitted, and any other messages available for transmission to the console. These characters can also cause application program actions and should be used with caution.

Paging Operations and Marking Input

Table 2-3 lists the keys used for output control and editing functions at terminals in all terminal classes. The table includes a terminal buffer clearance key. You normally clear the terminal buffer of a line mode or block mode terminal when:

Paging operations are expected during subsequent output.

The beginning of input must be distinguished from the end of the last output.

Paging operations, described in more detail in section 5, depend on one of the following conditions being in effect:

The console output mechanism must scroll all output. Scrolling occurs when a display fills with sequential output; the terminal places

each subsequent output line at the bottom of the display after moving existing lines up and removing the top line from the display.

The output mechanism must not contain any remnants of previous output or input.

When the terminal output mechanism does not perform scrolling, proper correlation of a page of output from the NPU to a single display of output on the console requires you to clear the screen of the output mechanism before NPU output begins. Screen clearance usually moves the device cursor (the input mechanism marker or carrier symbol) to the top left of the device output page area.

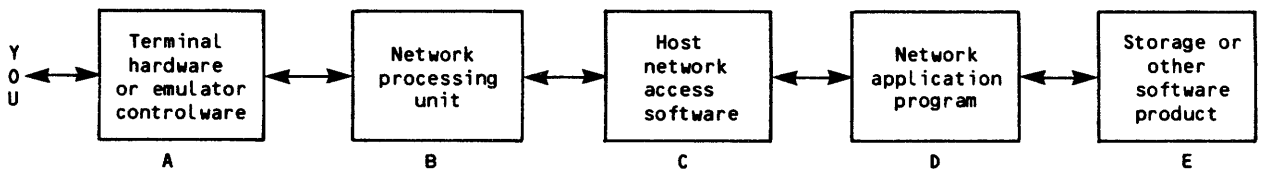
Block mode terminals and line mode terminals that do not scroll the terminal buffer usually distinguish the beginning of input based on a character symbol placed in the terminal buffer after output is completed. This symbol either requires terminal operator placement or is provided automatically by the terminal. When the symbol is absent from the terminal buffer, input is usually assumed to begin at the beginning of the buffer.

Terminals for which the beginning of input location can be tracked do not require entry of the symbol or clearance of the buffer. At other terminals, you can either explicitly enter the symbol that determines the beginning of input, or clear the buffer before typing the input. The two operations are equivalent. Failure to perform one of these operations when necessary can cause a portion of previous output or input to be included in the current input transmission.

DATA INPUT AND OUTPUT CONVERSIONS

One of the mysteries associated with the use of computers is why a character entered from the terminal is sometimes returned to the terminal as a different character, or else seems to disappear entirely. There are two separate but related reasons for this: character code translation, and character input stream editing.

When you press a keyboard key or create a pattern of paper tape or Hollerith card holes, the first of several sequential translation and editing operations begins. The following paragraphs trace a line of characters entered from a keyboard, through these operations, and back to the terminal as output. Figure 2-3 summarizes the steps involved.



Console Input Summary

Between A and B, terminal code set is used.
Between B and C, ASCII code set or terminal code set is used.
Between C and D, ASCII code set, display code set, or terminal code set is used.
Between D and E, ASCII code set, display code set, terminal code set, or application-defined code set is used.

Console Input Examples

- Normalized mode input to IAF from 2741
  Between A and B, EBCD; between B and D, ASCII; between D and E, 6-in-12-bit display code.
- Normalized mode input to RBF from HASP
  Between A and B, EBCDIC; between B and D, ASCII; between D and E, 6-bit display code.
- Transparent mode input to IAF from TTY
  Between A and B, ASCII with parity; between B and D, ASCII without parity, between D and E, binary data.

Batch Device Input Summary

Between you and A, binary data is used. Between A and B, terminal code set is used.
Between B and C, terminal code set or 6-bit display code is used.
Between C and E (D is bypassed), terminal code set or 6-bit display code is used.

Figure 2-3. Input and Output Code Conversion Points (Sheet 1 of 2)

Batch Device Input Examples

- Nontransparent input to RBF from 3780
  Between card reader and A, binary codes for 026 or 029 punches; between A and B, EBCDIC; between B and E, 6-bit display code.

- Network transparent input to RBF from HASP
  Between card reader and A, binary codes for any punches; between A and B, EBCDIC; between B and E, EBCDIC in 12-bit bytes.

Batch Device Output Summary

Between E and C (D is bypassed) binary data, ASCII in 12-bit bytes, or 6-bit display code is used.
Between C and B, binary data, ASCII in 12-bit bytes, or 6-bit display code is used.
Between B and A, binary data or terminal code set is used.

Batch Device Output Examples

- Nontransparent 95-character printer output from RBF to 714
  Between E and B, 12-bit ASCII; between B and A, ASCII.

- Nontransparent punch output from RBF to 3780
  Between E and B, 6-bit display code; between B and A, EBCDIC.

- Transparent plotter output from RBF to HASP
  Between E and A, 8-bit binary instruction codes.

Console Output Summary

Between E and D, application-defined code set is used.
Between D and C, 6-bit display code, ASCII, or terminal code set is used.
Between C and B, ASCII or terminal code set is used.
Between B and A, terminal code set is used.

Console Output Examples

- Normalized mode output from IAF to 2741
  Between E and D, 6-in-12-bit display code; between D and B, ASCII; between B and A, EBCD.

- Normalized mode output from RBF to HASP
  Between E and D, 6-bit display code; between D and B, ASCII; between B and A, EBCDIC.

- Transparent mode output from IAF to TTY
  Between E and D, binary data; between D and B, ASCII without parity; between B and A, ASCII with parity.

Figure 2-3. Input and Output Code Conversion Points (Sheet 2 of 2)

TERMINAL INPUT CONVERSIONS

At the moment a character becomes data, all correspondence between that byte of data and a character you recognize ceases. A character begins its journey through the network as binary digits in a pattern of varying length. The bit pattern created when you press a key is determined by the terminal manufacturer. The pattern created is called a terminal character code from the terminal's character code set. The correspondence between a given code and a given graphic or control character you recognize is simply a convention used by the terminal manufacturer.

A line of characters entered from the keyboard becomes a string of character codes during transmission from the terminal to the network processing unit. This string of codes corresponding to the characters you entered is usually prefixed and/or suffixed by other codes corresponding to control characters in the terminal's character code set. The prefixed and/or suffixed codes are called the communication line protocol envelope for the transmission.

The characters you entered have now undergone one translation and editing, at the point marked A in figure 2-3. Potentially, three more alterations to the characters can be performed (points B, C, and D in the figure) before the characters are acted upon as input data (point E in the figure).

NPU INPUT CONVERSIONS

When the character string reaches the network processing unit (point B in the figure), one of two actions is taken:

Normalized mode input is normalized and packed for transmission to the host computer.

Transparent mode input is packed for transmission to the host computer.

Normalized Mode

Normalized mode input is usually the initial mode of terminal servicing. If the terminal is being serviced in normalized mode, all transmissions from the terminal are manipulated so that they appear like entry of characters from any other terminal connected to the network. This manipulation is called normalization and occurs in stages.

First, each character received from the terminal is translated from the terminal code set used to represent it to a code for a comparable character in the ASCII character set. If no comparable character exists in the ASCII character set for a terminal code corresponding to a graphic character, the terminal character code is translated to the ASCII character for a blank. If no comparable character for a terminal code corresponding to a control character exists, the terminal character code is translated to the ASCII code for a null. These translations are summarized in appendix A.

At this stage, all normalized mode data is represented by a string of ASCII character codes for graphic and control characters. Next, those control codes associated by the network processing unit with line protocol functions and input editing mode are removed from the string of input characters. The control codes that have this association are determined by the line protocol the terminal uses. The line protocol, in turn, is determined by the network processing unit from the terminal class associated with the terminal.

When editing of the input character string is completed, the remaining ASCII character codes are sent to the network software resident in the host computer.

Transparent Mode

If the terminal is being serviced in transparent mode, its character strings bypass both the translation and editing steps. (Mode 4 terminal line protocol control character codes are deleted, however.) The terminal character codes reach the host software packed into bytes eight bits long (any unused bits at the upper or high order end of the byte are zero-filled).

HOST SOFTWARE INPUT CONVERSIONS

The character string has now reached point C in the figure and might have been translated and edited twice. The next operations performed on the data are determined by the application program with which the terminal is communicating.

The application program can request the network access software in the host to deliver the character string without additional modification; if the program does so, no translation or additional editing occurs at point C in the figure. This is always the case with data input from a terminal being serviced in transparent mode.

If the application program requests the operation, however, the host software at point C translates the ASCII character codes to codes for characters in the 6-bit display code set used by other host software. Because a maximum of 64 characters can be coded in the 6-bit display code set, translation of the ASCII characters from the 8-bit ASCII codes consists of translating all remaining ASCII control character codes to blanks and translating two different ASCII graphic character codes to the same display code graphic character code. The latter is usually described as folding the lowercase characters to the uppercase characters, or folding columns of the ASCII character set table in appendix A into one another.

Once the application program has received the character string of input, it potentially has been translated and edited three times. The application program can now perform additional translation and editing of the character strings before the data is actually processed.

Figure 2-3 contains several examples of the translations performed for the application programs described in other network product manuals. Whether or not an application program can properly handle transparent mode transmissions partially determines the fate of such input. Before using transparent mode for input, or before attempting to enter any character that cannot be represented directly in the display code character set, you should read the appropriate sections of the reference manual for the application program being used.

HOST SOFTWARE OUTPUT CONVERSIONS

Output from a network application program undergoes similar operations at each point in figure 2-3. If the application program does not generate the data itself, it obtains the data from another source (point E) and might have to translate the characters from codes used at point E to either display codes, terminal character codes, or ASCII character codes before sending the characters to the network access software. This translation (point D) depends solely on the application program.

If the application program transmits output in transparent mode, it must convert each character to output in the bit pattern of the terminal character code appropriate for the terminal destined to receive the character. No conversions can occur at points C or B for transparent mode output data.

If the application program transmits its output in normalized mode, it can select from two options:

- It can transmit in 6-bit display codes.
- It can transmit in ASCII character codes.

If it transmits in 6-bit display code characters, the host portion of the network software at point C

translates the character codes to ASCII character codes. Note that this translation allows output of a maximum of 64 different characters; only 64 characters can be represented in 6-bit display code, and the host's network supervision software cannot supply codes for characters it does not receive from the application program.

If the application program transmits its output in normalized mode but does not use display code, it must transmit the characters as ASCII codes. This option bypasses the translation performed at point C in the figure and preserves all graphic character codes and control codes that can be represented in the 128-character ASCII code set described in appendix A.

NPU OUTPUT CONVERSIONS

When the network processing unit receives strings of normalized mode output characters, it must translate those characters into terminal character set codes and provide an appropriate line protocol envelope for them. This translation, at point B in figure 2-3, is a mirror of the one performed at this point for input.

ASCII control character codes with no corresponding terminal character codes are converted to the terminal idle character (effectively becoming an ASCII null character). Other control character codes are converted to corresponding terminal control character codes.

ASCII graphic character codes are converted to terminal character codes for corresponding graphic characters. If no corresponding graphic character can be found, character set folding occurs as described in appendix A.

TERMINAL OUTPUT CONVERSIONS

When the characters of output finally reach the terminal at point A in the figure, they undergo one final translation. The terminal character codes representing the character string become whatever characters the terminal manufacturer has chosen to associate with the codes in that string. If output occurs on a terminal with a changeable typing element, the same character code can be externally represented by as many different characters as there are interchangeable typing elements for the terminal. The effect of this translation is most obvious when input data is stored while the terminal is using one typing element character set and then output while a different element (using a different character set) is in place.

A character of input is the same character on output only when all of the translations it undergoes en route are perfectly symmetrical. Such perfect symmetry exists for uppercase letters, digits, and a few other characters. Symmetry rarely exists for all characters in any terminal character set. This is why a character can be sent to the host as a ≠ but return as a quotation mark.

TERMINAL OPERATIONS 3

Section 1 indicated that the network software supports 18 classes of terminals, grouped into the following major terminal types:

Terminals on asynchronous communication lines, using either a teletypewriterlike communication protocol, such as the CDC 721 Viking Extended Display Terminal, or using IBM 2741 protocol.

Teletypewriterlike terminals, such as the CDC 721, using an X.25 packet-switching network for connection to the CDC network.

Terminals on synchronous communication lines using CDC mode 4 communication protocols, such as CDC 200 User Terminals (which use mode 4A protocol) and CDC 714 Conversational Display Terminals (which use mode 4C protocol).

Terminals on synchronous communication lines using variants of the HASP communication protocol, such as the CDC CYBER 18 using the COMM-18 emulation package.

Terminals on synchronous communication lines using either IBM 2780 or IBM 3780 terminal emulation protocol, such as a Wang system 30.

Terminals on synchronous communications lines using the IBM 3270 protocol.

Some of these terminals require you to load controlware before accessing the network; others do not. After access to the network, you can operate all these terminals in the general manner described in section 2. Special operating and loading requirements of selected terminals are described in the following subsections.

This section briefly supplies necessary information for the use of major terminal types and emulative equipment. You must obtain complete information for use of specific protocols or terminals from the appropriate equipment operation manual.

ASYNCHRONOUS TERMINALS

The asynchronous terminals supported as terminal classes 1 through 8 provide the most flexibility in use of the CDC network. These are the only terminals that can use all of the network software's features, and the only terminals for which the network software explicitly adapts its operations to accommodate paper tape operations.

PAPER TAPE OPERATIONS

There are three types of paper tape equipment:

Paper tape reader/punches and emulators (usually magnetic tape cassette units) that physically stop and start in response to control character codes received by the terminal.

Paper tape reader/punches that generate ASCII DC1 codes as X-ON commands and ASCII DC3 codes as X-OFF commands, or that respond to those characters when received. These mechanisms should not be identified to the network as paper tape units; instead, you should define them as keyboards with an IC command and/or OC command value of Y (refer to section 5).

Paper tape reader/punches and emulators (usually magnetic tape cassette units) that do not physically stop and start in response to control codes received by the terminal.

The network software only supports the first type as true paper tape equipment. If an ASCII DC3 code does not stop the paper tape reader, you should not indicate that your terminal has a paper tape input mechanism. In such cases, you should manually start and stop the reader or punch and your input should not contain X-OFF (DC3) codes.

Because the network software supports output as a logically separate operation from input, you can use paper tape for input only, or for output only. Commands are available to control either form of paper tape use. These commands are completely described in section 5 under the headings Selecting an Input Mechanism and Transmission Mode, and Selecting an Output Mechanism. This subsection presents the use of those commands in a functional context.

Paper Tape Reader/Punches

You can use paper tape for preparing input off-line (when time is not charged) and entering it on-line (when time is charged). This ensures accuracy and speed when most needed. An input tape can include programs, data, and commands. Accordingly, it is possible for the entire terminal operation, after login, to be run from paper tape.

This subsection describes use of a Model 33, 35, or 38 ASR Teletype (terminal class 1) with a paper tape punch and a paper tape reader. At these terminals, an X-OFF code on the tape stops the reader, and the ASCII code DC3 is the X-OFF code used by the reader. You need not manually start and stop paper tape units at such terminals; input you prepare for such devices can contain X-OFF codes.

The paper tape punch has one switch with several positions, or corresponding buttons. The switches or buttons are labeled and used as follows:

ON or $ROTR_{ON}$ or KT or TT_r

Press this button or place the switch in this position to turn the punch on.

OFF or K

Press this button or place the switch in this position to turn the punch off.

REL.

Press this button or place the switch in this position to release the feedwheel so that the tape can freely move through the punch head. Not all punches have a comparable control.

B.SP. or BKSP

Press this button or place the switch in this position to backspace the tape one row of holes. This button or switch position is used to make corrections when preparing a tape off-line (refer to the detailed description later in this subsection). On some models, the comparable control button is the keyboard key with this label or the label BACK SPACE.

The paper tape reader has one switch with several positions. The switches and buttons are labeled and used as follows:

START or TT_S

Press the button or push the switch to this position to start the reader. On some models, the switch snaps back to the AUTO or ON position and reading continues. On teletypewriter readers lacking an AUTO or ON position, you must manually restart reading each time it stops.

$TD_{CALL-IN}$ and TD_{ON}

Press these buttons to manually start tape reading.

STOP or TD_{OFF}

Move the switch to this position or press the button to stop reading immediately.

AUTO or ON

This switch position or button is used in conjunction with the input and processing of commands and data during input in paper tape normalized mode (this mode is explained later in this section). It allows the network software to turn the tape reader off and on so that processing of each command or line of data can be completed before the next one is entered.

FREE

Place the switch in this position to release the feedwheel so that the tape can freely slide in and out of the read head. Not all readers have a comparable control.

Preparing Paper Tape Input

Each line of input from paper tape in paper tape normalized mode must end with a particular sequence of characters. These are punched by keys on the teletypewriter keyboard.

The terminating characters used when punching paper tape are represented as follows:

[cr]

Message terminator and end-of-logical-line indicator, usually the carriage return character code. The carriage return code is generated by the RETURN key on all three Teletype models. It moves the carriage back to the beginning of the line and informs the NPU that this logical line of input is completed.

LF

Line feed character code. This character code is produced by the LINE FEED key on the keyboard. This advances the paper roller one line. During input in keyboard character mode, the NPU sends back a signal that initiates line feed after a carriage return; during input in tape normalized mode, this signal is not sent and is therefore required on tape to cause line feed on the output device.

X-OFF

End-of-line and end-of-block indicator, always the X-OFF tape unit control character code. The appearance of this character on the tape during reading is intended to turn the tape reader off. This character is punched by holding down the CTRL or CONTROL key and pressing the X-OFF or DC3 key. If your terminal keyboard has both a DC3 or X-OFF key and a key marked TAPE with a line through it (~~TAPE~~) (or only the latter key), do not use the latter key. That key does not generate the character code recognized by the network software for the X-OFF function.

RO(n)

Rubout or delete character code. This character is produced by the RUB OUT or DELETE key on the keyboard. This key punches a full row of holes. This row is interpreted as a null character by the network software; therefore, this character is used for spacing and overpunching errors. The n parameter specified by the user specifies the minimum number of times this character should be punched in sequence.

The following formats are used to transmit physical lines, single logical lines, or multiple logical lines. The left half of the column shows an example of the input, and the right half shows the terminating characters that immediately follow the last input character. Each line is terminated with three rubouts to provide separation from the next line. Although fewer rubouts might provide adequate separation, three is the recommended number.

The NPU punches three spacing characters at the end of each line when a tape is produced on-line. Input lines are blank-padded to an even number of characters on output.

A single physical line of input from a multiple physical line message would have a format similar to the following:

100 PROGRAM TEST (INPUT,OUTPUT) LF [cr] RO(3)

A single logical line, single transmission input message would have a format similar to the following:

12.44,18.31,29.08 [cr] LF X-OFF RO(3)

The X-OFF turns the reader off to allow this data to be processed before the next message is read.

A single logical line of input from a multiple line (typeahead) transmission would have a format similar to the following:

CATLIST,LO=F [cr] LF RO(3)

Omitting the X-OFF does not turn the reader off to allow processing of the command to be completed before the next logical line is read.

The following procedure is used to punch a tape off-line (when the terminal is not logged in):

1. Turn the teletypewriter on in local mode by pressing the LCL or LOCAL button.
2. Turn the paper tape punch on by pressing the KT button or the ON button located on the punch.
3. Prepare a tape leader of about 30 rubout characters by pressing the RUB OUT and repeat (REPT) keys together, by holding the DELETE key down, or by punching blank tape if the terminal has the capability.
4. With a pencil, trace an arrow above the punch output onto the tape. This identifies the start of the tape for reading. (You might put the tape in the reader backwards.)
5. Type in the input with appropriate control characters.
6. Add a trailer of about 30 rubouts or blank tape and tear the tape off.
7. Turn the paper tape punch off by pressing the K button or the OFF button located on the punch.
8. Press the CLR or CLEAR ALARM button to turn the teletypewriter off.

Beginning and Ending Paper Tape Operations

To read and process data and commands from tape, the tape reader sometimes must be turned off and on to allow time for processing. By entering tape input normalized mode, you enable the NPU to synchronize tape input with application program processing.

You place your terminal in this mode by entering a terminal definition command to change the input device to a paper tape reader (refer to section 5 for descriptions of these commands and defaults). Normally, this command appears as follows:

ESC IN=PT [cr]

In this command and the others described in this subsection, ESC identifies the ESC or ESCAPE key that generates an ASCII ESC control code.

The network acknowledges entry into this mode with the message:

IN ACCEPTED..

It then positions the carriage to the beginning of the line and performs two line feeds. If the reader switch is in the AUTO or ON position, reading begins immediately and automatically; if the reader is turned off, reading does not begin until it is manually initiated by momentarily moving the switch to START or pressing the $TD_{CALL-IN}$ and TD_{ON} buttons.

If you are using a Model 33 or 38, you exit from normalized paper tape input mode by entering a similar terminal definition command that changes the input device back to the keyboard. You enter:

ESC IN=KB [cr]

On the Model 35, you must press the K or KT button before you enter the command.

You can also define the paper tape punch as the output device using a terminal definition command. You do this by entering:

ESC OP=PT [cr]

If you are using a Model 33 or 38, you specify this command to set up the network software for outputting information to paper tape. This mode provides the network-recognized X-OFF characters when punching the tape. On the Model 35, you must press the KT or TT_R button before you enter the command.

If you are using a Model 33 or 38, you exit paper tape output mode by entering a command to change the output device back to a printer. You do this by entering:

ESC OP=PR [cr]

On the Model 35, you must press the K button before you enter this command. A similar command, described in section 5, permits you to change the output device back to a display.

Creating Paper Tape Output

To punch a tape with information from the host computer, use the following procedure:

1. If not logged in, log in.
2. If not in tape output character mode, type:

   ESC OP=PT [cr]

3. Type in, but do not transmit, an application program command to begin output of the information from the host.
4. Turn the punch on.
5. Manually enter a leader of about 30 null characters or rubouts.

6. Press [cr]. The information is listed and punched simultaneously. The NPU adds the appropriate control characters at the end of each line.

7. Manually enter a trailer of about 30 null characters or rubouts.

8. Turn the punch off.

9. Enter the cancel-input character (section 2) and press [cr]. This avoids sending the trailer rubouts to the application program as an empty logical line.

While punching a tape off-line, you can make corrections by backspacing over the incorrect punch (use the B.SP. or BKSP button or switch position on the punch) and punching a rubout over the error (use the RUB OUT or DELETE key on the keyboard). Then punch in the correct character.

You can also make corrections by using the backspace key as in normal, on-line operation using the keyboard for input. However, the error and/or any overtyping appear in the printout. You can retype and repunch an erroneous line. Although the erroneous and correct versions appear in printout, the NPU transmits only the occurrence of the undeleted line.

Examples of Paper Tape Use

The following portions of terminal sessions illustrate paper tape use for input and output while communicating with the network application program IAF. The standard version of the Interactive Facility program released by CDC is separately described in the NOS reference manual set.

For the first example, suppose you have punched an input tape containing multiple logical lines off-line. This tape is shown in figure 3-1. The left half of the figure gives the printed copy produced on the teletypewriter printer as the tape was punched; the right half shows the unprinted terminating characters that were also entered on the tape.

	RO(30)		
100 LET FACTOR=1	[cr]	LF	RO(3)
110 INPUT N	[cr]	LF	RO(3)
120 FOR I=1 TO N	[cr]	LF	RO(3)
130 LET FACTOR=FACTOR*I	[cr]	LF	RO(3)
140 PRINT FACTOR	[cr]	LF	RO(3)
150 NEXT I	[cr]	LF	RO(3)
160 END	[cr]	LF	RO(30)

Figure 3-1. Off-line Tape for Example 1

In this example, you use paper tape only for program input, and only to supplement additional logical lines typed in before and during reading of the paper tape. Interleaved tape and keyboard operation is always possible.

To demonstrate the use of the tape shown in figure 3-1, assume that you are logged in to the IAF NULL subsystem and using a Model 38. The tape is positioned in the reader and you set the reader switch to ON.

The printout shown in figure 3-2 results from the interaction of you and the tape with the network software and IAF. The right half of the figure explains the actions occurring in the listing. The printing character % has been substituted for the nonprinting character ESC throughout the session so that the listing is easier to read.

The terms normalized mode and keyboard mode are explained in sections 2 and 5, rather than within the figure where they occur. These terms are not significant in the context of this subsection. The characters identified as NUL(2) correspond to two ASCII null character codes and occur on the tape as sets of spacer holes only.

In the second example, the input tape contains a program; commands to execute, list, modify, and save that program; as well as the input data used by that program. You punch the complete tape off-line as shown in figure 3-3. The left half of the figure gives the printed copy that is produced as you punch the tape; the right half of the figure shows the terminating characters entered at the end of each line but not printed.

To process the information on this tape, you log into IAF and position the tape in the Model 38 reader. You set the reader switch to ON, type in the command ESC IN=PT, and press [cr] to start the reading of tape. The remainder of the action occurs without your intervention.

Figure 3-4 shows the printout of the execution of the tape. The right half of the figure contains explanations of the actions.

TERMINAL FEATURE DEPENDENCIES

Documents written for terminal users identify two types of characters:

Control characters that are not representable on an output device

Graphic characters that are representable on an output device.

On some asynchronous terminals, such as the CDC 751, it is possible to symbolically represent all graphic and control characters on the output device. A graphic character cannot therefore be defined solely as a displayable or printable character.

References to graphic characters in this manual are to alphabetic, numeric, punctuational, or notational symbols. Graphic characters are those in the ASCII character set table in appendix A that are not represented by mnemonic abbreviations for functions.

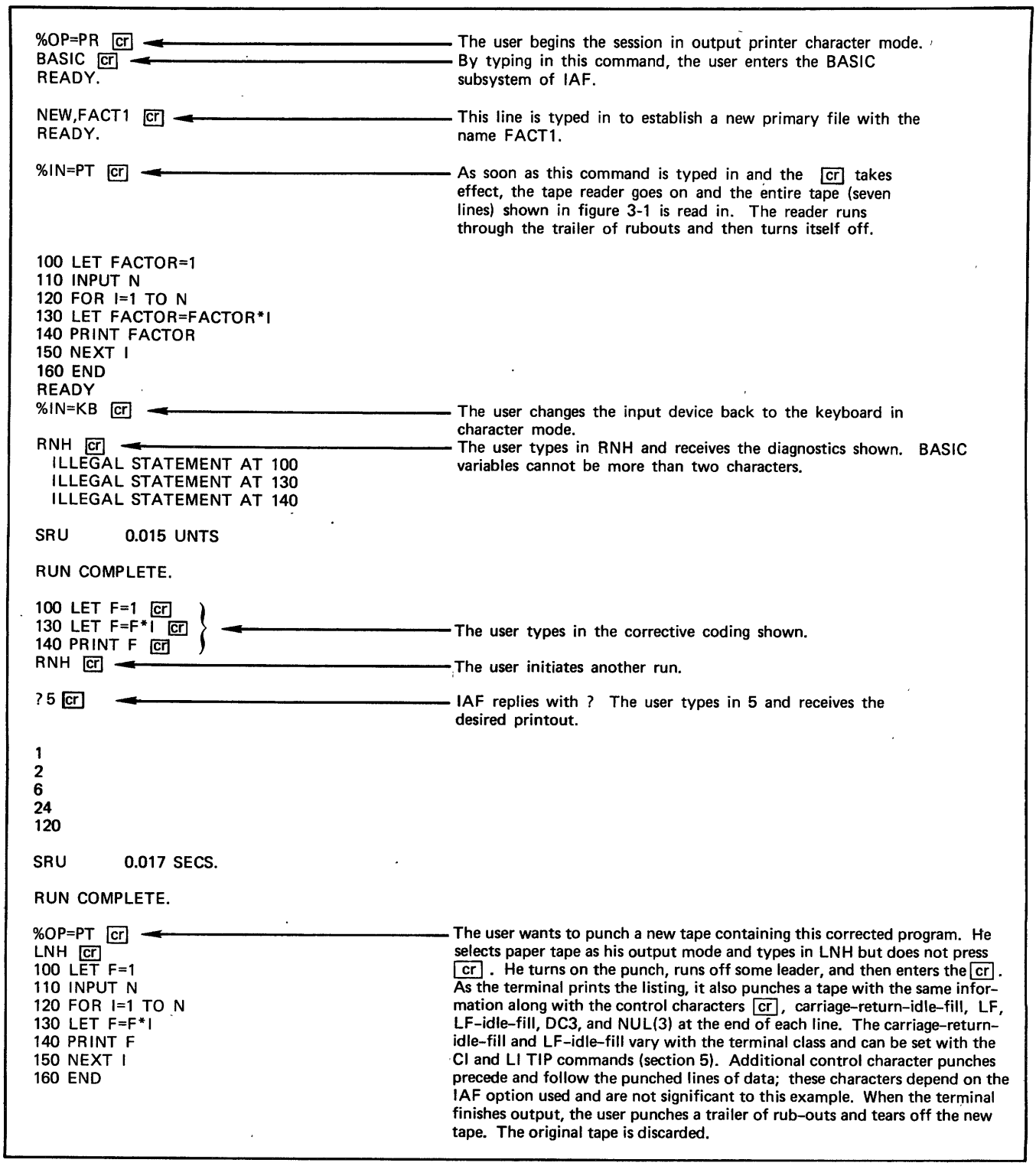


Figure 3-2. Example 1 Terminal Session

	RO(30)			
FTNTS	cr	LF	X-OFF	RO(3)
NEW,DEMO	cr	LF	X-OFF	RO(3)
100 PROGRAM DEMO(INPUT,OUTPUT)	cr	LF	RO(3)	
110 DIMENSION II(5)	cr	LF	RO(3)	
120 READ 10, (II(J),J=1,5)	cr	LF	RO(3)	
130 10 FORMAT(5I5)	cr	LF	RO(3)	
140 PRINT 20, (II(J),J=1,5)	cr	LF	RO(3)	
150 20 FORMAT(5I10)	cr	LF	RO(3)	
160 END	cr	LF	RO(3)	
RNH	cr	LF	X-OFF	RO(3)
11111222223333344444455555	cr	LF	X-OFF	RO(3)
SAVE,DEMO=TAPE1	cr	LF	X-OFF	RO(3)
135 DO 1 I=1,5	cr	LF	RO(3)	
137 II(I)=II(I)+4444	cr	LF	RO(3)	
139 1 CONTINUE	cr	LF	RO(3)	
RESEQ,100,5	cr	LF	X-OFF	RO(3)
LNH	cr	LF	X-OFF	RO(3)
RNH	cr	LF	X-OFF	RO(3)
11111222223333344444455555	cr	LF	X-OFF	RO(3)
REPLACE,DEMO=TAPE1	cr	LF	X-OFF	RO(30)

Figure 3-3. Off-Line Tape for Example 2

```
FTNTS <-------- When this line is read, the reader stops. As soon as FTNTS is
                established as the current subsystem, the prompt from IAF turns
                the reader back on.

NEW,DEMO <----- When this line is read, the reader stops. After a new primary
                file called DEMO is established, the next prompt from IAF
                turns the reader back on.

100 PROGRAM DEMO(INPUT,OUTPUT) <-- The seven lines of the program are read in without interruption.
110 DIMENSION II(5)
120 READ 10, (II(J),J=1,5)
130 10 FORMAT (5I5)
140 PRINT 20, (II(J),J=1,5)
150 20 FORMAT (5I10)
160 END

RNH <---------- The command to run the program is read in and the reader
                turned off.

? 11111222223333344444455555 <-- Processing reaches the READ statement (line 120) and IAF
                prints ?. This causes the NPU to turn the reader back on
                and the line of data is read in. Then the reader is turned off
                to allow data processing.

11111  22222  33333  44444  55555 <-- Resulting printout at the terminal.
END.

SRU        0.130 UNTS.

RUN COMPLETE.

SAVE,DEMO=TAPE1 <-- This command is read in and the reader turned off. A copy
                of the program DEMO is made an indirect access permanent
                file with the name TAPE1. Then the reader is turned back
                on by the next prompt output.

135 DO 1 I=1,5 <--- These three lines of modification are read in from the tape
137 II(I)=II(I)+4444 without interruption.
139 1 CONTINUE

RESEQ,100,5 <---- The RESEQ command is read in and the reader turned off.
                When resequencing is accomplished according to specifications,
                the reader is turned back on by the next prompt output.
```

Figure 3-4. Example 2 Terminal Session (Sheet 1 of 2)

```
SRU        0.576 UNTS.

LNH
```

This command is read from tape and the tape reader turned off. Then IAF lists the primary file which contains the above modifications in resequenced format. When the listing is completed, the reader is turned on again.

```
00100 PROGRAM DEMO(INPUT,OUTPUT)
00105 DIMENSION II(5)
00110 READ 10,(II(J),J=1,5)
00115 10 FORMAT(5I5)
00120 DO 1 I=1,5
00125 II(I)=II(I)+4444
00130 1 CONTINUE
00135 PRINT 20,(II(J),J=1,5)
00140 20 FORMAT (5I10)
00145 END

RNH
```

This command is read in, the reader turned off, and the modified program run. IAF prints the ? which turns the reader on. The line of data is read in from the tape and the reader is turned off. The data is processed and results printed.

```
? 111112222233333444445555 5


15555  26666  37777  48888  59999
END.

SRU        1.185 UNTS.

RUN COMPLETE.

REPLACE,DEMO=TAPE1
```

This last command is read in and the reader turned off. A copy of this revised version of DEMO replaces the old one that was made an indirect access permanent file under the name TAPE1. The revised file is now the indirect access permanent file that is referenced by the name TAPE1.

The reader is turned back on. It runs through the trailer of rubouts and then turns itself off.

```
ESC IN=KB [cr]
```

At this point, the user has no more tapes to run, and exits tape input character mode by entering the TIP command. The tape reader is then manually turned off by moving the switch to the STOP position.

Figure 3-4. Example 2 Terminal Session (Sheet 2 of 2)

Using Slave Devices

The network does not support more than one device on an asynchronous communication line. Only one terminal console can be serviced on a line. Any other terminals or devices using the communication line must operate as slave devices to the terminal console.

Such slave devices often include a printer. When you use a printer as a slave device to a console for which a display is assumed to be the output mechanism, output characters can be lost during physical repositioning of the printer mechanism. When this occurs, you can use the terminal definition commands for carriage return and line feed idle character insertion (the CI and LI commands, described in section 5) to correct the situation.

Using Devices Controlled by Data Codes

If you use a tape cassette for input or output at an asynchronous terminal, you should consider the mechanism to be a paper tape emulator only if the terminal does not associate inappropriate functions with the receipt of character codes corresponding to the ASCII characters DC1 and DC3. The same restriction is true for use of input control (see the IC command in section 5).

For example, a Teletype Model 40 reacts to receipt of the DC3 code by switching from batch mode (block mode transmission) to conversational mode (character mode transmission). Defining the cassette mechanism as a paper tape for output purposes would cause the network to switch the input operation of the keyboard and cassette unintentionally. To avoid

this situation, you should regard such a cassette mechanism as a keyboard, display, or printer mechanism only.

Terminals such as the Hazeltine 2000 permit ASCII control character codes in output data so that display functions can be deliberately altered. You should not place the code for the ASCII unit separator character US in normalized mode output for such purposes. The unit separator character is used to delimit output lines within blocks sent from the host computer. This character is removed from normalized mode output by the NPU. If the unit separator character appears anywhere within an output transmission, it causes either a single-spaced carriage return and line feed operation at the terminal or interpretation of the following character as a format effector.

Using 2741 Terminals

Terminals such as the IBM 2741 which use a shift code to distinguish between uppercase and lowercase characters during input, rather than distinct character codes, require operator action to preserve the distinction between input lines. Because each line of input is edited and transmitted separately by the network processing unit, the shift to uppercase occurring in one line is not preserved for the processing of the next line. Each input line is assumed to begin in lowercase, so you must press or relock the shift key again at the beginning of each line when the distinction between uppercase and lowercase is significant in the input data. Case distinction during output occurs without operator intervention.

All 2741-compatible terminals supported by a CDC network should have a transmit-interrupt feature to permit entry of multiple input lines without interleaved output.

IBM-2741-compatible terminals can use the ATTN key to perform a line feed function when the special editing feature described in sections 2 and 5 is in use. If such terminals possess an LF key, that key can also be used for this function. The ATTN key should never be used solely to reposition paper on the output mechanism.

The ATTN key must be pressed before you can enter one of the following network software function characters:

cancel-input
abort-block
user-break-1
user-break-2
network control

These characters are defined by the respective CN, AB, B1, B2, and CT commands described in section 5.

Most asynchronous terminals need an automatic recognition procedure to identify themselves to the network. After you perform this procedure as described in section 4, hit the RETURN key to lock the keyboard so that output can occur. Terminals on fixed configuration lines do not require the automatic recognition procedure, but you must hit the RETURN key to lock the keyboard so that the 2741 can receive output. Unless the network software recognizes that the keyboard is locked, it will not send output.

Timing is important. If you press the RETURN key before the 2741 is physically connected to the network, the keyboard locks but the 2741 will not receive the output that begins the login sequence. The actual time needed for terminal connection varies based on the configuration and the network activity. The average time to wait before pressing the RETURN key is 3 to 5 seconds. However, if you press the RETURN key and the keyboard locks but no login sequence is displayed, first switch the 2741 to local mode; then switch it back to on-line mode. This unlocks the keyboard and allows you to press the RETURN key again.

Coping With Terminal Installation Options

Terminals that have a transmission key marked NL or NEW LINE might generate an NL code when the key is pressed. The NL code is interpreted as a carriage return code followed by a line feed code when received by such terminals. The network sees an NL code as a line feed code but interprets it as a carriage return if the terminal uses an appropriate terminal class.

Some terminals allow selection among alternative sequences of carriage return and line feed codes transmitted by NL or NEW LINE keys. The NPU reacts to the sequence of codes received; the operation associated with such keys in this manual might not be the operation occurring at your terminal because of this variation in code sequence.

The network does not support use of answerback options by asynchronous terminals. Character codes generated by such device options are not equivalent to completing the login procedure described in section 4. It is possible for the site to configure a terminal so that the login procedure is automatically completed for it, but this operation is performed in the host computer without any action by you.

OUTPUT FORMATTING

A network application program can format each line of its output to an asynchronous console by beginning the line with an ASCII character identified as a format effector. The resulting operations are shown in table 3-1.

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
1	blank	Space 1 line before output.	Does not matter	Yes	CR	CR
				No	CR, LF	CR, LF
	0	Space 2 lines before output.	Does not matter	Yes	CR, LF	CR, LF
				No	CR, 2LF	CR, 2LF
	-	Space 3 lines before out put.	Does not matter	Yes	CR, 2LF	CR, 2LF
				No	CR, 3LF	CR, 3LF
	+	Position to start of current line before output.	Does not matter	Does not matter	CR	CR
	*	Position to top of form or home cursor before output.††	Yes	Yes	CR, 5LF	CR, 5LF
				No	CR, 6LF	CR, 6LF
			No	Does not matter	Calculated by TIP	
	1	Position to top of form or home cursor and clear screen before output.††	Yes	Yes	CR, LF	CR, 5LF
				No	CR, 6LF	CR, 6LF
			No	Does not matter	Calculated by TIP	
	,	Do not change position before output.	Does not matter	Does not matter	None	None
	.	Space 1 line after output.	Does not matter	Does not matter	CR,LF	CR,LF, DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Does not matter	CR	CR, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes	CR	CR
				No	CR, LF	CR, LF
2	blank	Space 1 line before output.	Does not matter	Yes	CR	CR
				No	CR, LF	CR, LF
	0	Space 2 lines before output.	Does not matter	Yes	CR, LF	CR, LF
				No	CR, 2LF	CR, 2LF
	-	Space 3 lines before output.	Does not matter	Yes	CR, 2LF	CR, 2LF
				No	CR, 3LF	CR, 3LF
	+	Position to start of current line before output.	Does not matter	Does not matter	CR	CR
	*	Position to top of form or home cursor before output.††	Does not matter	Does not matter	EM	EM
	1	Position to top of form or home cursor and clear screen before output; delay 100 millisecends before further output.††	Does not matter	Does not matter	EM, CAN	EM, CAN

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES (Contd)

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
2 (Contd)	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	CR, LF	CR, LF DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	CR	CR, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF
3	blank	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF
	0	Space 2 lines before output.	Does not matter	Yes No	CR, LF CR, 2LF	CR, LF CR, 2LF
	-	Space 3 lines before output.	Does not matter	Yes No	CR, 2LF CR, 3LF	CR, 2LF CR, 3LF
	+	Position to start of current line before output.	Does not matter	Yes or No	CR	CR
	*	Position to top of form or home cursor before output.††	Does not matter	Yes or No	ESC, FF	FF
	1	Position to top of form or home cursor and clear screen before output.††	Does not matter	Yes or No	ESC, FF	FF
	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	CR, LF	CR, LF DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	CR	CR, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES (Contd)

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
4†††	blank	Space 1 line before output.	Does not matter	Yes No	None NL	N/A
	0	Space 2 lines before output.	Does not matter	Yes No	NL 2NL	N/A
	-	Space 3 lines before output.	Does not matter	Yes No	2NL 3NL	N/A
	+	Position to start of current line before output.	Does not matter	Yes or No	nBS n is calculated by TIP from current position	N/A
	*	Position to top of form or home cursor before output.††	Yes	Yes No	5NL 6NL	N/A
			No	Yes or No	nNL n is calculated by TIP from current position	N/A
	1	Position to top of form or home cursor and clear screen before output.††	Yes	Yes No	5NL 6NL	N/A
			No	Yes or No	nNL n is calculated by TIP from current position	N/A
	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	NL	NL
	/	Position to start of current line after output.	Does not matter	Yes or No	nBS n is calculated by TIP from current position	nBS
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	None NL	None NL
5	blank	Space 1 line before output.	Does not matter	Yes No	None LF	None LF
	0	Space 2 lines before output.	Does not matter	Yes No	LF 2LF	LF 2LF
	-	Space 3 lines before output.	Does not matter	Yes No	2LF 3LF	2LF 3LF
	+	Position to start of current line before output.	Does not matter	Yes or No	ESC, G	ESC, G

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES (Contd)

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
5 (Contd)	*	Position to top of form or home cursor before output.††	Does not matter	Yes or No	ESC, H	ESC, H
	1	Position to top of form or home cursor and clear screen before output.††	Does not matter	Yes or No	ESC, R	ESC, R
	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	LF	LF, DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	ESC, G	ESC, G, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	None LF	None LF
6	blank	Space 1 line before output.	Does not matter	Yes or No	CR	CR
	0	Space 2 lines before output.	Does not matter	Yes No	CR 2CR	CR 2CR
	-	Space 3 lines before output.	Does not matter	Yes No	2CR 3CR	2CR 3CR
	+	Position to start of current line before output.	Does not matter	Yes or No	None	None
	*	Position to top of form or home cursor before output.††	Does not matter	Yes or No	DC2	DC2
	1	Position to top of form or home cursor and clear screen before output.††	Does not matter	Yes or No	FS	FS
	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	CR	CR, DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	None	DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes or No	CR	CR

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES (Contd)

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
7	blank	Space 1 line before output.	Does not matter	Yes No	CR CR,LF	CR CR, LF
	0	Space 2 lines before output.	Does not matter	Yes No	CR, LF CR, 2LF	CR, LF CR, 2LF
	-	Space 3 lines before output.	Does not matter	Yes No	CR, 2LF CR, 3LF	CR, 2LF CR, 3LF
	+	Position to start of current line before output.	Does not matter	Yes or No	CR	CR
	*	Position to top of form or home cursor before output.††	Does not matter	Yes or No	ESC, [, H	ESC, [, H
	1	Position to top of form or home cursor and clear screen before output.††	Does not matter	Yes or No	ESC, [, H, ESC, [, J	ESC, [, H, ESC, [, J
	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	CR, LF	CR, LF DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	CR	CR, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF
8	blank	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF
	0	Space 2 lines before output.	Does not matter	Yes No	CR, LF CR, 2LF	CR, LF CR, 2LF
	-	Space 3 lines before output.	Does not matter	Yes No	CR, 2LF CR, 3LF	CR, 2LF CR, 3LF
	+	Position to start of current line before output.	Does not matter	Yes or No	CR	CR
	*	Position to top of form or home cursor before output.	Does not matter	Yes or No	ESC, FF	ESC, FF
	1	Position to top of form or home cursor and clear screen before output; delay 1 second before further output.	Does not matter	Yes or No	ESC, FF	ESC, FF

TABLE 3-1. FORMAT EFFECTOR OPERATIONS FOR ASYNCHRONOUS AND X.25 CONSOLES (Contd)

Terminal Class	Format Effector	General Physical Operation	Is Infinite Page Length Declared?	Does Output Follow Previous Input	Code Substituted on Output Mechanism	
					Display or Printer	Paper Tape†
8 (Contd)	,	Do not change position before output.	Does not matter	Yes or No	None	None
	.	Space 1 line after output.	Does not matter	Yes or No	CR, LF	CR, LF, DC3, 3NUL
	/	Position to start of current line after output.	Does not matter	Yes or No	CR	CR, DC3, 3NUL
	Any other ASCII character	Space 1 line before output.	Does not matter	Yes No	CR CR, LF	CR CR, LF

†Paper tape column does not apply to X.25 devices.

††If these format effectors appear anywhere in a data block other than in the first or last line of the data block, the TIP can perform a page-wait operation before the format effector operation. (See section 5.)

†††X.25 devices cannot belong to terminal class 4.

Conversion of characters in your output to format effector characters can be performed by the network application program communicating with the terminal. If the network application program does not perform such conversions, all such characters are output at the terminal and all lines are single-spaced. The network application program might provide these characters without having them supplied in your output. This is the most common situation for output to asynchronous terminals.

The application program can:

Suppress the use of format effectors completely

Provide its own format effectors

Expect your output to supply all format effectors

If format effectors are suppressed, all output is single-spaced. If the application supplies its own format effectors, you can use the first character of each of your data's physical lines as a format effector by prefixing the character with an ASCII unit separator (US) character. If the application requires your data to contain format effectors, the US prefix is unnecessary. When a character of your data is used as a format effector, that character is transformed as shown in the table and is deleted from the output sent to the terminal.

The format effector characters for clear screen and home cursor operations (* and 1) receive special treatment by the Terminal Interface Program when it is performing a page wait function for the terminal. (See the PG command in section 5.) If these characters are encountered when the TIP has output only part of a page, the TIP pauses for terminal operator acknowledgment of the partial page. When acknowledgment occurs, the format effector functions are performed and output continues automatically. This pause occurs without application program action or knowledge.

TERMINALS USING X.25 NETWORKS FOR ACCESS

The asynchronous terminals supported as terminal classes 1 through 3 and 5 through 8 can also use a CDC network through an X.25 packet-switching network. These terminals address the X.25 network through a packet assembly/disassembly (PAD) service. The PAD takes care of most terminal dependencies and determines many terminal operating features seen by the CDC network.

TERMINAL FEATURE DEPENDENCIES

Documents written for terminal users identify two types of characters:

Control characters that are not representable on an output device

Graphic characters that are representable on an output device

On some asynchronous terminals, such as the CDC 751, it is possible to represent all graphic and control characters symbolically on the output device. A graphic character cannot therefore be defined solely as a displayable or printable character.

References to graphic characters in this manual are to alphabetic, numeric, punctuational, or notational symbols. Graphic characters are those in the ASCII character set table in appendix A that are not represented by mnemonic abbreviations for functions.

Using Slave Devices

The network does not support more than one device on an X.25 communication line virtual circuit. Only one terminal console can be serviced on a circuit. Any other terminals or devices using the communication line must operate as slave devices to the terminal console on the same circuit, or as separate terminal consoles on separate circuits.

Such slave devices often include a printer. When you use a printer as a slave device to a console for which a display is assumed to be the output mechanism, output characters can be lost during physical repositioning of the printer mechanism. When this occurs, you can use the terminal definition commands for carriage return and line feed idle character insertion (the CI and LI commands, described in section 5) to correct the situation.

Using Devices Controlled by Data Codes

Terminals such as the Hazeltine 2000 permit ASCII control character codes in output data so that display functions can be deliberately altered. You should not place the code for the ASCII unit separator character US in normalized mode output for such purposes. The unit separator character is used to delimit output lines within blocks sent from the host computer. This character is removed from normalized mode output by the NPU. If the unit separator character appears anywhere within an output transmission, it causes either a single-spaced carriage return and line feed operation at the terminal or interpretation of the following character as a format effector.

Using the PAD Service

One of the data forwarding parameters recognized by your terminal's PAD service should be the same as the end-of-line character recognized by the CDC network. PAD services usually forward data after a carriage return code is entered. Carriage return is therefore the default end-of-line indicator character for all terminal classes and corresponds to a value of 2 for the setting of PAD parameter 3.

If the PAD allows you to define more than one forwarding signal, make the second signal the end-of-block indicator recognized by the CDC network.

If your terminal's PAD is not configured this way, the CDC network will pass the second PAD forwarding character to the host application as part of the input data. If your terminal's PAD is configured this way, the CDC network removes the second PAD forwarding character from the data.

If you cannot alter your terminal's PAD parameter settings, you should use the EL terminal definition command described in section 5 to alter the way the network processes your input. Setting the end-of-line character equal to your PAD forwarding character defines your terminal as a line mode device.

The PAD can usually perform any cursor positioning response (using PAD parameter 13) you need when you end a logical line or a block. If you do not ask the CDC network to do cursor positioning (the default choice for the CP command described in section 5), you save the time and cost of transmitting a packet through the X.25 network each time you end a line or block.

Some PAD services support DC3 (X-OFF) and DC1 (X-ON) codes as device input control codes. You can use the IC terminal definition command described in section 5 to set PAD parameter 5 to select or deselect this option.

Some PAD services recognize an ASCII DC3 (X-OFF) code from a terminal as a control signal to interrupt output. A subsequent ASCII DC1 (X-ON) code resumes output from the PAD. You can use the OC terminal definition command described in section 5 to set PAD parameter 12 to select or deselect this option.

If you need the PAD to provide character echoing, you should use the EP terminal definition command described in section 5 to set PAD parameter 2 to select or deselect this option.

Coping With Terminal Installation Options

Terminals that have a transmission key marked NL or NEW LINE might generate an NL code when the key is pressed. The NL code is interpreted as a carriage return code followed by a line feed code when received by such terminals. The network sees an NL code as a line feed code but interprets it as a carriage return if the terminal uses an appropriate terminal class.

Some terminals allow selection among alternative sequences of carriage return and line feed codes transmitted by NL or NEW LINE keys. The NPU reacts to the sequence of codes received; the operation associated with such keys in this manual might not be the operation occurring at your terminal because of this variation in code sequence.

The network does not support use of answerback options by asynchronous terminals. Character codes generated by such device options are not equivalent to completing the login procedure described in section 4. It is possible for the site to configure a terminal so that the login procedure is automatically completed for it, but this operation is performed in the host computer without any action by you.

OUTPUT FORMATTING

A network application program can format each line of its output to an X.25 console by beginning the line with an ASCII character identified as a format effector. The format effectors allowed are the same as those used for asynchronous consoles directly connected to a CDC network. The resulting operations are shown in table 3-1.

The application program can:

Suppress the use of format effectors completely

Provide its own format effectors

Expect your output to supply all format effectors

If format effectors are suppressed, all output is single-spaced. If the application supplies its own format effectors, you can use the first character of each of your data's physical lines as a format effector by prefixing the character with an ASCII unit separator (US) character. If the application requires your data to contain format effectors, the US prefix is unnecessary. When a character of your data is used as a format effector, that character is transformed as shown in the table and is deleted from the output sent to the terminal.

The format effector characters for clear screen and home cursor operations (* and 1) receive special treatment by the Terminal Interface Program when it is performing a page wait function for the terminal. (See the PG command in section 5.) If these characters are encountered when the TIP has output only part of a page, the TIP pauses for terminal operator acknowledgment of the partial page. When acknowledgment occurs, the format effector functions are performed and output continues automatically. This pause occurs without application program action or knowledge.

MODE 4 TERMINALS

Mode 4A terminals are divided into two groups:

Those requiring switch setting and loading of controlware for proper operation, such as the CDC 73x-12 and CDC 734 (terminal class 15).

Those requiring only switch settings for proper operation, such as the 200 User Terminal 217 display console (terminal class 10) and CDC 214 (supported under terminal class 10).

Mode 4C terminals, such as the CDC 711 (terminal class 12) or CDC 714 (terminal classes 11 and 13), do not require special preparation before accessing the network.

MODE 4A SWITCH SETTINGS

The ATTENDED/UNATTENDED switch on the CDC 200 User Terminal 217 Series or 214 Series display keyboard must be set to ATTENDED. Use of mode 4A terminals in unattended mode is not described in this manual; attended mode is the recommended way to use mode 4A terminals with a CDC network. The exception to this rule is the CYBER 18 COMM-18 emulator, which requires unattended mode selection through the /EXT command to a 200 User Terminal simulator.

The LINE/BLOCK switch on mode 4A consoles should be set to LINE. Use of the BLOCK setting for mode 4A terminals is not completely described in this manual; the LINE setting is recommended when using mode 4A terminals with CDC host software.

Detailed descriptions of terminal controls and operation are provided in the respective terminal operation manuals.

MODE 4A LOADING

You must load a 73x Remote Batch Terminal station prior to placing it on-line if the station was placed off-line or powered-down since the last on-line usage. All of these terminals, except the 734, store their executable controlware on a microdrum; the loading procedure stores the controlware in memory. You load the 734 Remote Batch Terminal from a card deck.

731-12 and 732-12 Remote Batch Terminals

You must load the 731-12 and 732-12 Remote Batch Terminals with the mode 4A emulation parameters. The loading process allows you to accept or alter the terminal's operating characteristics.

The selections you make among the options offered should match the description of a terminal on the communication line you are going to use. You must describe the terminal to itself as possessing the characteristics described to the network software by the network administrator when the network configuration file was created. Contact a site administrator if in doubt about what options to select.

The modes of operation you specify as selections should include line mode operation and either 029 or 026 Hollerith card reading translations. These choices also include the site address and either External BCD (EXT BCD) or ASCII as the terminal character code set (the option of Internal BCD, or INT BCD, is not supported by the network).

Once you have made all desired selections, loading is completed. The procedures described in section 4 can now be performed.

TERMINAL FEATURE DEPENDENCIES

Mode 4A terminals are all CDC 200 User Terminal emulators. Such terminals must contain the following devices:

An interactive console

A card reader for batch input

A line printer for batch output

Mode 4C terminals can contain:

One or more interactive consoles

Zero, one, or more line printers

One console is required as an owning console for the batch devices.

Choosing Punched Card Pattern Translations

Almost all mode 4A terminals have a way you can select proper input translation of Hollerith cards punched with either 026 or 029 codes. This method is the only method that can be used to switch from one punch code translation to the other. CDC local batch card readers and other remote batch stations allow a card that switches between these code translations as part of the card deck input; such cards in data do not have any effect when the data is read in from a mode 4A terminal card reader.

The code translations performed from 026 or 029 Hollerith cards to terminal character codes (and character codes sent to the host) vary from emulator to emulator. The most common translations are those shown in appendix A.

Positioning the Cursor

The network software attempts to place the display cursor and send index (or start index) symbols at the left and right margins, respectively (when appropriate for the terminal class), after each normalized mode output or input transmission, based on the assumption of line mode operation. All normalized mode output to the console and all responses to normalized mode input transmissions are blank-padded only when necessary to correctly position the cursor for subsequent line mode input.

If you use the terminal in block mode, correct identification of the beginning of console input probably will not occur. If no method exists for choosing between line mode and block mode, you can safely assume line mode is used.

The blank padding used to position the cursor can cause an extra blank line on the screen at some terminals. This line is not part of the input or output data.

Access and Disconnection Procedures

A mode 4 terminal sharing a communication line with one or more terminals in the same or another cluster can use slightly different access and disconnection procedures than those used by other terminals. During the procedure to access the network, only one mode 4C console in each cluster needs to perform the automatic recognition procedure.

After logout from the network, a mode 4 device is not physically disconnected from the network until all mode 4 devices in the cluster are logged out. Disconnection occurs only if the communication line is a dialup line; terminals on hardwired communication lines are never disconnected. Until physical disconnection occurs, you can initiate new host connection and login procedures, as described in section 4, without repeating any of the preliminary access procedures.

Using the Devices

You cannot simultaneously use the console and batch devices of a mode 4A terminal. You can switch the console from communication with one network application program to communication with another, as described in section 4. You cannot switch the batch devices from communication with a network application program unless the program itself performs the switch.

OUTPUT FORMATTING

A network application program can format each line of its output to a mode 4 console by beginning the line with an ASCII character identified as a format effector. The resulting display operations are shown in table 3-2.

The application program can:

Suppress the use of format effectors completely

Provide its own format effectors

Expect your output to supply all format effectors

If format effectors are suppressed, all output is single-spaced. If the application supplies its own format effectors, you can use the first character of each of your data's physical lines as a format effector by prefixing the character with an ASCII unit separator (US) character. If the application requires your data to contain format effectors, the US prefix is unnecessary. When a character of your data is used as a format effector, that character is transformed as shown in the table and is deleted from the output sent to the terminal.

The format effector characters for clear screen and home cursor operations (* and 1) receive special treatment by the Terminal Interface Program when it is performing a page wait function for the terminal. (See the PG command in section 5.) If these characters are encountered when the TIP has output only part of a page, the TIP pauses for terminal operator acknowledgment of the partial page. When acknowledgment occurs, the format effector functions are performed and output continues automatically. This pause occurs without application program action or knowledge.

A network application program can format each line of its output to a mode 4 printer by beginning each print line with an ASCII character identified as a carriage control character. The resulting operations are shown in table 3-3.

Conversion of characters in your output to carriage control characters is controlled by the network application program communicating with the terminal. If the application program does not require such conversions, all such characters are output at the terminal and all lines are single-spaced.

TABLE 3-2. FORMAT EFFECTOR OPERATIONS FOR MODE 4 CONSOLES

Terminal Class	Format Effector	General Physical Operation†	
		Before Output††	After Output††
10 thru 13, and 15	blank	None.	Space 1 line.
	0	Space 1 line.	Space 1 line.
	-	Space 2 lines.	Space 1 line.
	+	None.	Space 1 line.
	*	Position to top of form or home cursor.†††	Space 1 line.
	1	Position to top of form or home cursor and clear screen.	Space 1 line.
	.	None.	Space 1 line.
	/	None.	Space 1 line.
	Any other ASCII character§	None.	Space 1 line.

†No direct correspondence to code substituted on output device can be made. Code used for implementation depends on placement of message blocks within a transmission.

††After each input and output line, the Terminal Interface Program returns the cursor to the beginning (left) margin of the next line.

†††If these format effectors appear anywhere in a data block other than in the first or last line of the data block, the TIP can perform a page-wait operation before the format effector operation. (See section 5.)

§Sent to the terminal.

TABLE 3-3. CARRIAGE CONTROL OPERATIONS FOR MODE 4 PRINTERS

ASCII Character	General Physical Operation	
	Before Printing	After Printing
Blank	Space one line.	None.
0	Space two lines.	None.
-	Space three lines.	None.
+	None.	None.
1	Space to top of next page.	None.
PM	None.	None.
Q	Space one line.	None.
R	Space one line.	None.
S	Space one line.	None.
T	Space one line.	None.

HASP WORKSTATION TERMINALS

HASP workstations supported by a CDC network comprise an operator's console and an associated group of batch devices. Each device within the workstation is separately addressed by its device type and input or output stream number during access to the network.

WORKSTATION SETUP BEFORE ACCESS

A HASP workstation can only access the CDC network by using a communication line explicitly configured to support the batch devices used by the workstation. The devices used at the workstation should agree in number and type with those expected to use the communication line.

For example, a HASP workstation with three card readers, a line printer, and two card punches should only access the network via a communication line configured to support three card readers, a line printer, and two card punches. Otherwise some of its batch devices will not be operable.

The CDC network can support HASP workstations comprising the following types of devices:

One operator console

Zero to seven card readers

Zero to seven line printers

Zero to seven card punches or plotters

If the workstation uses a device for input or output that is not included in this list, the device must be configured to appear as one of these devices.

Using Stream Numbers

All devices used must be configured with an input or output stream number unique for the device within the group of devices of the same type. That is, two card punches using stream number 2 are not permitted, but one card punch and one line printer each using stream number 2 are permitted. A plotter and a card punch cannot share the same stream number.

Stream numbers identify the devices within a workstation to the workstation. Device ordinal numbers identify the devices within a workstation to the host software. When the site administrator creates the network configuration file used by the host software, each HASP device allowed to use a communication line is assigned a device ordinal number.

The host software and the workstation must both identify a specific device by the same number. For example, a HASP workstation with two printers assigned to stream numbers 2 and 3 cannot use a communication line configured for a HASP workstation with two printers assigned device ordinal numbers 1 and 4, even though the quantity of devices and of streams is the same.

Some workstations provide a switch or software for assigning or changing the stream number associated with a device. If your workstation is one of these types and is on a dialup line, do the following:

Contact the site administrator and find out what device ordinal numbers are used for the communication line you are supposed to use.

Use the workstation switch or software to assign those device ordinal numbers as the stream numbers for the terminals within the workstation.

If more than one dialup line is available, always use the line for which the current workstation stream number assignments are valid device ordinal number assignments, or reassign the stream numbers accordingly.

If a stream number assignment method is not available to you, you must determine the numbers used by the workstation and consult the site administrator for the dialup number of the correct communication line to be used. Hardwired terminals are normally correctly configured at the host computer before communication with the network begins.

TERMINAL FEATURE DEPENDENCIES

You must be careful when using the operating guide for a HASP workstation. Many such guides discuss operations outside of the workstation proper as operations performed by an IBM host computer. In a CDC network, many of these operations are performed in the network processing units without the knowledge of the CDC host.

Using IBM Host Features

Use of a HASP workstation with an IBM host computer requires transmission of a block of signon information to the host. This signon information can either be generated automatically within the workstation or be entered manually through the workstation console or a card reader.

The method used to provide the signon block information depends on the workstation used. Documentation of the signon procedure usually states that signon to the host computer is necessary and that a response to a successful signon procedure comes from the host computer. In a CDC network, the signon block is not processed this way.

The CDC network requires a login procedure containing four parameters to be completed for the HASP workstation console terminal. This login procedure can be completed automatically as described in section 4 or be performed manually by you. The CDC login procedure is independent of the signon block procedure.

To provide compatibility with the procedures used by sites with HASP workstations connected to an IBM host computer, the CDC network processes the signon block in the manner described in section 4. Regardless of the method used to enter the signon information, the signon block is removed from the workstation input by the network processing unit.

If the information contained in the signon block is not meaningful to the network software, any information that is transmitted by the workstation is acknowledged as a valid and successful signon by the network processing unit. However, the network processing unit does not establish communication with the CDC host computer to begin the CDC login procedure when an automatic recognition /*CONFIG statement is separately required.

Similar processing is performed for the signoff procedure used by IBM host computers. Transmission of a signoff block, such as that generated by the reading of a /*SIGNOFF card, causes acknowledgment of the block, but no logical or physical disconnection of the workstation. To disconnect the workstation from the CDC network, you must perform the disconnection procedures described in section 4.

Using the Devices

After login to the host computer is completed, you can simultaneously use the console and batch devices of the HASP workstation. You can switch the console from communication with one network application program to communication with another, as described in section 4. You cannot switch the batch devices from communication with a network application program unless the program itself performs the switch.

OUTPUT FORMATTING

A network application program can format each line of its output to a HASP console by beginning the line with an ASCII character identified as a format effector. The resulting display operations are shown in table 3-4.

TABLE 3-4. FORMAT EFFECTOR OPERATIONS FOR HASP CONSOLES

Terminal Class	Format Effector	General Physical Operation†	
		Before Output	After Output
9 and 14	0	Space 1 line.	Space 1 line.
	-	Space 2 lines.	Space 1 line.
	Any other ASCII character	None.	Space 1 line.

†No direct correspondence to code substituted on output device can be made. Code used for implementation depends on placement of message blocks within a transmission.

The application program can:

Suppress the use of format effectors completely

Provide its own format effectors

Expect your output to supply all format effectors

If format effectors are suppressed, all output is single-spaced. If the application supplies its own format effectors, you can use the first character of each of your data's physical lines as a format effector by prefixing the character with an ASCII unit separator (US) character. If the application requires your data to contain format effectors, the US prefix is unnecessary. When a character of your data is used as a format effector, that character is transformed as shown in the table and is deleted from the output sent to the terminal.

A network application program can format each line of its output to a HASP printer terminal by beginning each print line with an ASCII character identified as a carriage control character. The resulting operations are shown in table 3-5.

Conversion of characters in your output to carriage control characters is controlled by the application program. If the application program does not require such conversions, all such characters are output at the terminal and all lines are single-spaced.

Using Terminal or Network Transparent Mode

Some HASP workstations use emulators that support commands to perform transparent terminal data transmissions to or from devices. These commands preserve character codes normally associated with data link escape functions from treatment as function codes by the workstation during transmission. The workstation commands have no effect on the use of the CDC network feature for console transparent mode transmissions, as described in section 2. Transparent terminal data transmission is described in section 6.

The CDC network does not permit network transparent mode transmissions to or from the console of a HASP workstation. Batch transparent card input data is supported as described in section 6; the EBCDIC character codes indicated in appendix A are stored in host files. If your host does not supply software to restructure and interpret such data, you should not use the transparent network data feature.

Data transmitted to a plotter is always transparent in the sense of not being converted to character codes; such data is treated as binary punch data and is presumed to be instructions for plotter operation transmitted in 6-bit or 8-bit bytes.

BISYNCHRONOUS 2780 OR 3780 TERMINALS

IBM 2780 or 3780 terminals supported by a CDC network comprise a simulated operator's console and associated batch devices.

TERMINAL SETUP BEFORE ACCESS

A 2780 or 3780 terminal can only access the CDC network by using a communication line configured to support the batch devices used by the terminal. The devices used at the terminal should agree in type with those expected to use the communication line.

For example, a 3780 terminal with a card punch should only access the network via a communication line configured to support a card reader, a line printer, and a card punch. Unless such a communication line is used, data cannot be sent to the card punch.

Each terminal can contain:

A card reader

A line printer

A simulated operator's console (using the card reader in place of a keyboard and the line printer in place of a display)

An optional card punch.

The first two devices are required. The simulated console always exists. Only the card punch can be addressed separately by its device selection code during access to the network. If the terminal uses a device for input or output that is not included in this list, the device must be configured to appear as one of these devices.

Using Punch Selection Codes

If your terminal has a card punch, the selection code used for that device must be the same as the code defined for the card punch supported by the network on the communication line in use. Contact your site administrator if you are not sure the correct code is used.

TABLE 3-5. CARRIAGE CONTROL OPERATIONS FOR HASP PRINTERS

ASCII Character	General Physical Operation		ASCII Character	General Physical Operation	
	Before Printing†	After Printing		Before Printing†	After Printing
Blank	Space one line.	None.	Q	Line not printed.	None.
0	Space two lines.	None.	R	Line not printed.	None.
-	Space three lines.	None.	S	Line not printed.	None.
+	None.	None.	T	Line not printed.	None.
/	Space one line.	None.	A	None.	Space to top of next page (skip to controller channel 1).
1	Space to top of next page (skip to channel 1).	None.	B	None.	Space to bottom of current page (skip to controller channel 12).
2	Space to bottom of current page (skip to controller channel 12).	None.	C	None.	Skip to controller channel 6.
3	Skip to controller channel 6.	None.	D	None.	Skip to controller channel 5.
4	Skip to controller channel 5.	None.	E	None.	Skip to controller channel 4.
5	Skip to controller channel 4.	None.	F	None.	Skip to controller channel 3.
6	Skip to controller channel 3.	None.	G	None.	Skip to controller channel 2.
7	Skip to controller channel 2.	None.	H	None.	Skip to controller channel 11.
8	Skip to controller channel 11.	None.	I	None.	Skip to controller channel 7.
9	Skip to controller channel 7.	None.	J	None.	Skip to controller channel 8.
X	Skip to controller channel 8.	None.	K	None.	Skip to controller channel 9.
Y	Skip to controller channel 9.	None.	L	None.	Skip to controller channel 10.
Z	Skip to controller channel 10.	None.			
PM	None.	None.			

†Not all HASP emulators can perform these operations.

Using 2780 Terminals

Some 2780 emulators allow a variation in the maximum number of records transmitted per terminal transmission block. The number of records per block your terminal transmits must be within the range allowed for a terminal on the communication line you are using.

TERMINAL FEATURE DEPENDENCIES

You must be careful when using the operating guide for a 2780 or 3780 terminal. Many such guides discuss operations outside of the terminal proper as operations performed by the IBM host computer. In a CDC network, many of these operations are performed in the network processing units without the knowledge of the host.

Using IBM Host Features

Use of a 2780 or 3780 terminal with an IBM host computer requires transmission of a block of signon information to the host. This signon information can either be generated automatically within the terminal or be entered manually through a card reader.

The method used to provide the signon information depends on the terminal used. Documentation of the signon procedure usually states that signon to the host computer is necessary and that a response to a successful signon procedure comes from the host computer. In a CDC network, signon is also processed this way.

The CDC network requires a login procedure containing four parameters to be completed for the 2780 or 3780 terminal console. This login procedure can be completed automatically as described in section 4 or be performed manually by you. The CDC login procedure is identical to the signon block procedure; only the information used differs.

Some IBM hosts require terminal configuration information as part of the signon procedure. In a CDC network, such information is processed separately before login, using the /*CONFIG statement described in section 4.

No processing is performed for the signoff procedure used by IBM host computers. Transmission of a /*SIGNOFF card causes acknowledgment of the card, but no logical or physical disconnection of the terminal. To disconnect the terminal from a host or the CDC network, you must perform the disconnection procedures described in section 4.

Using the Devices

After login to the host is completed, you cannot simultaneously use the console and batch devices of a 2780 or 3780 terminal. Only one of the following devices can be used at a time:

Card reader as a console keyboard

Card reader as a batch input device

Line printer as a console printer

Line printer as a batch output device

Card punch as a batch output device

Batch device operation is essentially continuous. Batch or console input always receives priority over output. Input is not interrupted for output; however, output can be interrupted for input. Interrupted output resumes after an end-of-transmission is received at the end of the input; the use of end-of-transmission is discussed below.

These rules mean that console input can occur only between batch input file transmissions. Console output can occur whenever batch output at the printer stops.

You can enter any command normally transmitted from a console. However, you must prefix all such commands with the characters /* . These prefixing characters are not transmitted to the host. For example, if you respond to the host login prompt of FAMILY: with the entry /*SYSSVL, the host receives a family name entry of SYSSVL.

You can switch the console from communication with one network application program to communication with another, as described in section 4. You cannot switch the batch devices from communication with a network application program unless the program itself performs the switch.

Using Terminal or Network Transparent Mode

Some 2780 or 3780 terminals perform terminal transparent mode transmissions to or from batch devices. This mode preserves character codes normally associated with data link escape functions from treatment as function codes by the terminal during transmission.

Some emulators have commands that can be entered from a console to perform such transparent mode transmissions. The emulator commands have no effect on the use of the CDC network feature for console transparent mode transmissions, as described in section 2. The NPU does not support network transparent mode transmissions to or from the console of a 2780 or 3780 workstation.

Batch transparent card input data is supported as described in section 6; the EBCDIC character codes indicated in appendix A are stored in host files. If your host does not supply software to restructure and interpret such data, you should not use the transparent network transmission feature.

Using End-of-Text and End-of-Transmission

When the card reader hopper of an IBM 2780 or 3780 becomes empty and the end-of-file key or switch is enabled, the terminal generates an end-of-transmission (EOT) code. It can then receive output from the network. Otherwise, an empty hopper generates only an end-of-text (ETX) code. These conventions allow you to read in more than one card deck without waiting for output to be completed.

Some 2780 or 3780 emulators also allow you to separate multiple input transmissions by either an ETX code or an ETX followed by an EOT code and a delay of varying length before the next input transmission begins.

The ETX-only option is adequate to separate batch input transmissions. However, the EOT option is preferrable to separate console transmissions so that the network has time to return responses from itself or the host before subsequent entries occur.

For example, a Wang Model 30 system allows documents (files) to be queued for transmission in the same 3780 "session" (separated by ETX codes only), or in separate 3780 "sessions" (separated by an EOT as well). The EOT option is necessary after sending a terminal definition command, /*CONFIG card, or login

parameter. Unless the EOT option is used, subsequent input is discarded (typeahead is not supported in those situations).

Coping With Transmission Blocks

Although transmission block size is not significant for some 3780 emulators, others require the network to tailor the output transmission block size to the terminal's hardware requirements. For example, 3780 console and printer output sent to a Wang Model 30 is actually written to that system's hard disk. If the transmission block size used is significantly larger than the disk sector size (255 characters), the Wang must discard part of each block because it cannot convert the output to its internal format and write the data on the disk before the next downline block arrives. Successful use of such terminals requires the network administrator to configure them with an appropriate transmission block size.

Some 2780 emulators allow from one to seven records in each output transmission block. Other 2780 emulators require a fixed number of records in output transmission blocks (two records per block is common). The number of records allowed by your terminal should agree with the number the network administrator defines for the terminal.

OUTPUT FORMATTING

A network application program can format each line of its output to a 2780 or 3780 console by beginning the line with an ASCII character identified as a format effector. The resulting display operations are shown in table 3-6.

The application program can:

Suppress the use of format effectors completely

Provide its own format effectors

Expect your output to supply all format effectors

If format effectors are suppressed, all output is single-spaced. If the application supplies its own format effectors, you can use the first character of each of your data's physical lines as a format effector by prefixing the character with an ASCII unit separator (US) character. If the application requires your data to contain format effectors, the US prefix is unnecessary. When a character of your data is used as a format effector, that character is transformed as shown in the table and is deleted from the output sent to the terminal.

A network application program can format each line of its output to a 2780 or 3780 printer by beginning each print line with an ASCII character identified as a carriage control character. The resulting operations are shown in table 3-7.

Conversion of characters in your output to carriage control characters is controlled by the application program. If the application program does not require such conversions, all such characters are output on the device and all lines are single-spaced.

3270 TERMINALS

The 3270 terminals supported by a CDC network comprise a console and a printer.

TERMINAL FEATURE DEPENDENCIES

Up to 32 controllers (clusters) on a line where each cluster can have a maximum of 32 devices (display stations and/or line printers). EBCDIC is the only code set supported. Card reader support (magnetic - id card) is implicitly available in transparent mode.

AUTO-RECOGNITION

Auto-recognition of 3270 terminals is not performed, therefore dial up terminals must dial into synchronous ports supporting 3270s only. The line definition for 3270 ports must include all cluster and terminal addresses that may connect to the port.

OUTPUT FORMATTING

A network application program can format each line of its output to a 3270 console by beginning the line with a character identified as a format effector. The resulting display operations are shown in table 3-8. A network application program can format each line of its output to a 3270 printer terminal by beginning each print line with a character identified as a carriage control character. The resulting operations are shown in table 3-9.

TABLE 3-6. FORMAT EFFECTOR OPERATIONS FOR 2780 AND 3780 BISYNCHRONOUS CONSOLES

Terminal Class	Format Effector	General Physical Operation†	
		Before Output	After Output
16 and 17	Any ASCII character	Before the first line of the message, generate the prefix text ***CONSOLE MESSAGE Before the subsequent lines of the message, do nothing.	Space 1 line. Space 1 line.

†No direct correspondence to code substituted on output device can be made. Code used for implementation depends on placement of message blocks within a transmission.

TABLE 3-7. CARRIAGE CONTROL OPERATIONS FOR 2780 AND 3780 PRINTERS

ASCII Character	General Physical Operation	
	Before Printing	After Printing†
Blank	Space one line.	None.
0	Space two lines.	None.
-	Space three lines.	None.
+	Space one line on 2780, or no action on 3780.	None.
1	Space to top of next page (skip to channel 1).	None.
2	Space one line on 2780, space to bottom of of current page (skip to controller channel 12) on 3780.	None.
3	Skip to controller channel 6.	None.
4	Skip to controller channel 5.	None.
5	Skip to controller channel 4.	None.
6	Skip to controller channel 3.	None.
7	Skip to controller channel 2.	None.
8	Space one line on 2780, skip to controller channel 11 on 3780.	None.
9	Skip to controller channel 7.	None.
X	Space one line on 2780, skip to controller channel 8 on 3780.	None.
Y	Space one line on 2780, skip to controller channel 9 on 3780.	None.
Z	Space one line on 2780, skip to controller channel 10 on 3780.	None.

TABLE 3-7. CARRIAGE CONTROL OPERATIONS FOR 2780 AND 3780 PRINTERS (Contd)

ASCII Character	General Physical Operation	
	Before Printing	After Printing†
PM	None.	None.
Q	Line not printed.	None.
R	Line not printed.	None.
S	Line not printed.	None.
T	Line not printed.	None.
A	None.	Space to top of next page (skip to controller channel 1).
B	Space one line on 2780, no action on 3780.	Space one line on 2780, space to bottom of current page (skip to controller channel 12) on 3780.
C	None.	Skip to controller channel 6.
D	None.	Skip to controller channel 5.
E	None.	Skip to controller channel 4.
F	None.	Skip to controller channel 3.
G	None.	Skip to controller channel 2.
H	Space one line on 2780, no action on 3780.	No action on 2780, skip to controller channel 11 on 3780.
I	None.	Skip to controller channel 7.
J	None.	Skip to controller channel 8.
K	Space one line on 2780, no action on 3780.	No action on 2780, skip to controller channel 9 on 3780.
L	Space one line on 2780, no action on 3780.	No action on 2780, skip to controller channel 10 on 3780.
Any other character	Space 1 line.	None.

†Not all emulators can perform these operations.

TABLE 3-8. FORMAT EFFECTOR OPERATIONS FOR THE 3270 BISYNCHRONOUS TERMINAL

Terminal Class	Format Effector	General Physical Operation	
		Before Output	After Output
18	Blank	None.	Space one line.
	0	Space one line.	Space one line.
	-	Space two lines.	Space one line.
	+	None.	Space one line.
	*	None.	Space one line.
	1	None.	Space one line.
	'	None.	Space one line.
	.	None.	Space one line.
	/	None.	Space one line.
	Others	None.	Space one line.

TABLE 3-9. CARRIAGE CONTROL OPERATIONS FOR 3270 BATCH PRINTERS

Terminal Class	Format Effector	Action	
		Before Printing	After Printing
18	Blank	Space one line.	None.
	0	Space two lines.	None.
	-	Space three lines.	None.
	+	None.	None.
	1	Space to top of next page.	None.
	Others	Space one line.	None.

ACCESS AND DISCONNECTION 4

This section describes procedures to access a host computer system and to use CDC network software when you need help that the host system cannot supply. Each procedure corresponds to a step in getting from the bottom of figure 4-1 to the top.

Reaching a network application program in a host involves eight distinct steps. Each step is always taken when a terminal accesses the network. However, you might not have to perform all of the procedures. Your network's administrators can make many of these procedures unnecessary by having the software take the step for you.

The following procedures are described in separate subsections:

Gathering information (step 1)

Setting up your terminal (step 2)

Connecting your terminal to the network (step 3)

Identifying your terminal to the network (step 4)

Selecting the host that runs the software you need (step 5)

Connecting your terminal to that host (step 6)

Identifying yourself to that host (step 7)

Connecting your terminal to the network application program that provides the services you need (step 8)

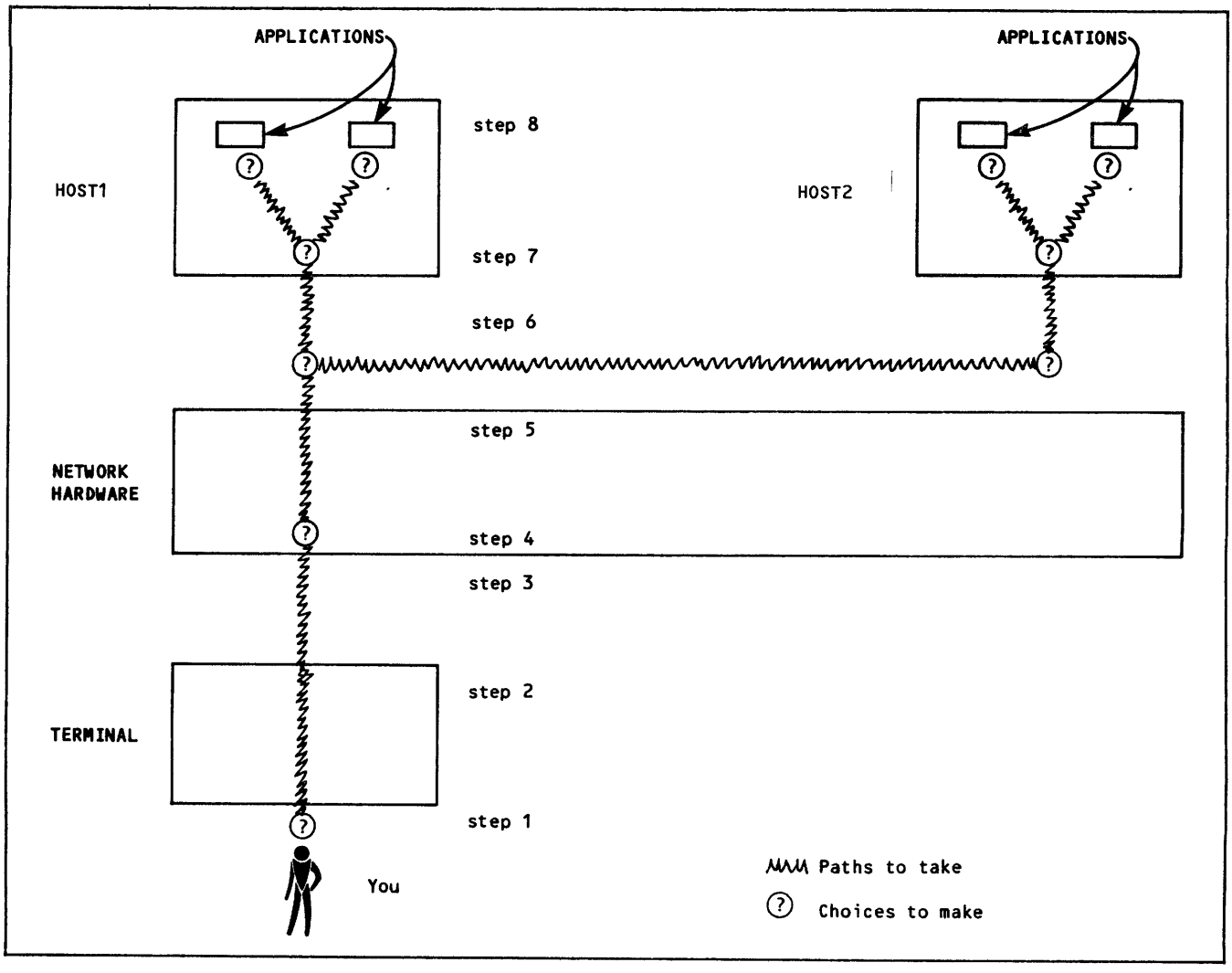


Figure 4-1. Steps in Accessing Host Software

GATHERING INFORMATION

The following questions can help you determine which of the subsequent procedures are necessary. Answering these questions is step 1, as shown in figure 4-2.

If you know the answers, or can get the answers from people at your site, the last paragraph of text following each question will tell you which questions and the corresponding subsections of this book you can skip. If you cannot answer a question, read all of the subsections; each procedure description contains hints for answering the questions.

1. Is the terminal hardwired, or is it a dial-up terminal?

Sections 1 and 3 can help you answer this question. If your terminal is hardwired, answer questions 2 through 5 and 7 through 10 below. Skip the subsection called Connecting Your Terminal.

2. Is your terminal configured for automatic recognition?

Your site can ask the network software to determine certain things about your terminal when the terminal joins the network. This process is called automatic recognition. (Terminals using X.25 packet-switching networks cannot use automatic recognition.)

If your terminal is configured for automatic recognition, you must take an extra step each time you connect your terminal to the network. If your terminal is not configured that way, answer questions 4 through 10 below. Skip the subsection called Identifying Your Terminal to the Network.

3. What protocol does your terminal use?

Section 3 explains protocols briefly, and section 6 provides more detail on how the network supports them. If your terminal operating manual does not tell you what protocol your terminal uses, question

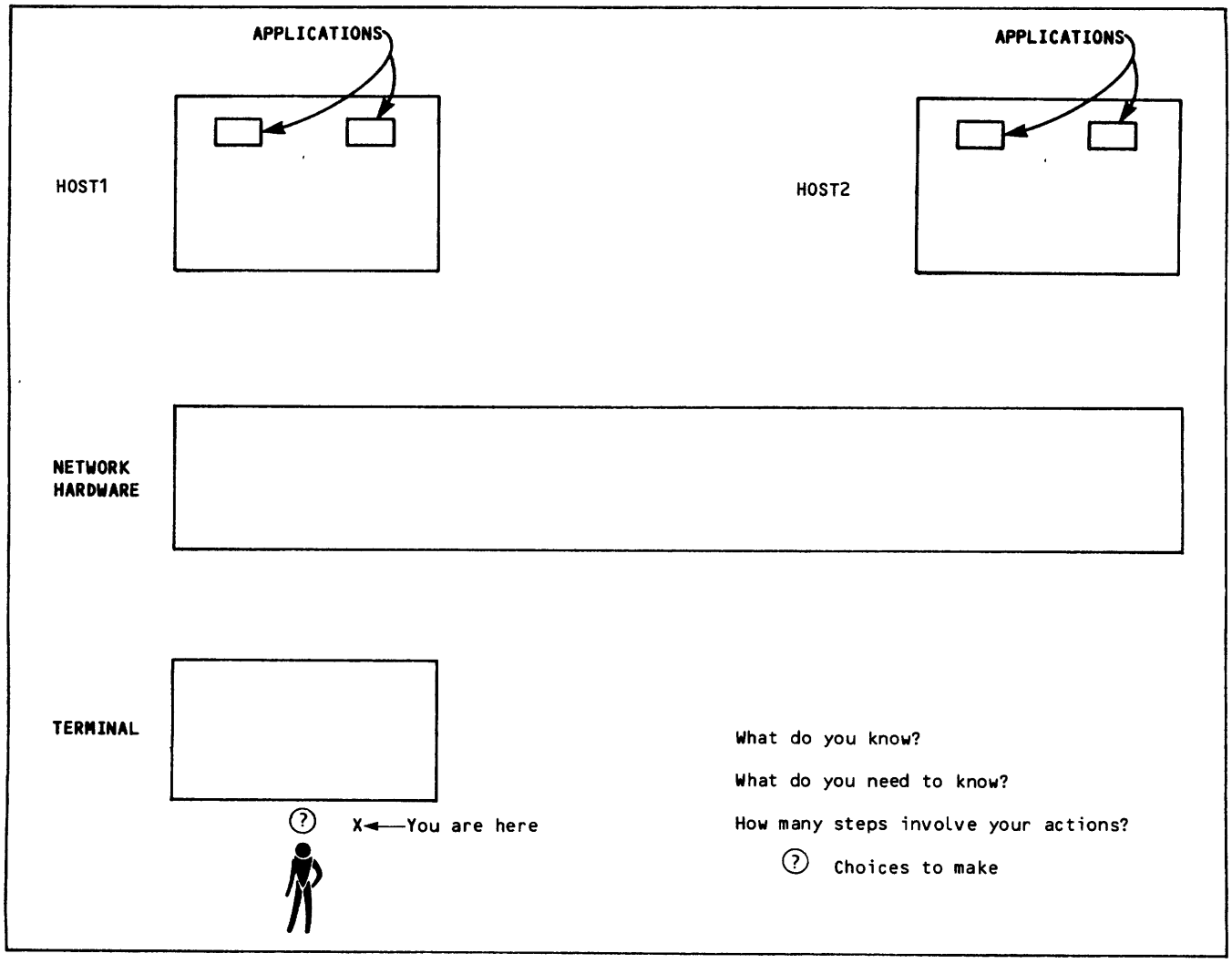


Figure 4-2. Step 1, Gathering Information

4 and the terminal classes shown in section 5 might help you answer this question. If your terminal is not configured for automatic recognition, answer questions 5 through 10 below. Skip the subsection called Identifying Your Terminal to the Network.

4. What class of terminal is it?

Many different kinds of terminals exist. The network software assigns terminals it supports to classes, as follows:

Class	Protocol
1 thru 8	asynchronous or X.25 asynchronous
9 and 14	HASP bisynchronous
10 and 15	CDC mode 4A synchronous
11 thru 13	CDC mode 4C synchronous
16 and 17	2780 and 3780 bisynchronous
18	3270 bisynchronous
other	site-defined

If you know what the terminal class is, that will tell you the protocol.

Terminals in classes 1 through 8 sometimes are called interactive terminals or consoles. A console has an input and output device that can be used for dialog with the network. Most consoles use a television-like screen or a typewriter-like mechanism to convey output information. Almost all consoles have a keyboard for entering information to the network. Some consoles might also use mechanisms such as magnetic tape cassettes or paper tape reader/punches for input or output.

Terminals in classes 9 through 18 sometimes are called remote batch terminals or remote job entry terminals. Such terminals have an interactive console and can also have batch devices, as described in section 1.

Figure 4-3 shows a console with a display screen and a keyboard. Figure 4-4 shows a console with a printer and a keyboard. Figure 4-5 shows a remote batch terminal with a console and a batch printer device.

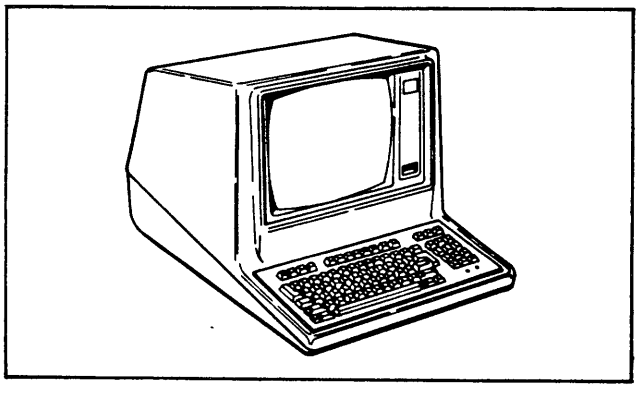


Figure 4-3. Interactive Display Console

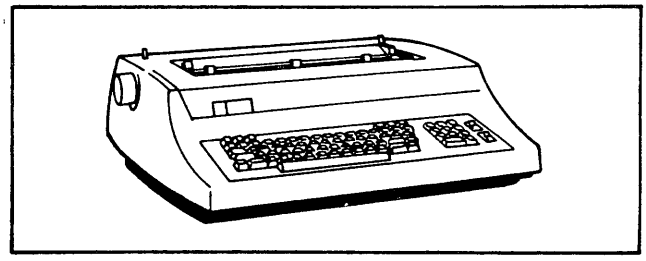


Figure 4-4. Interactive Printer Console

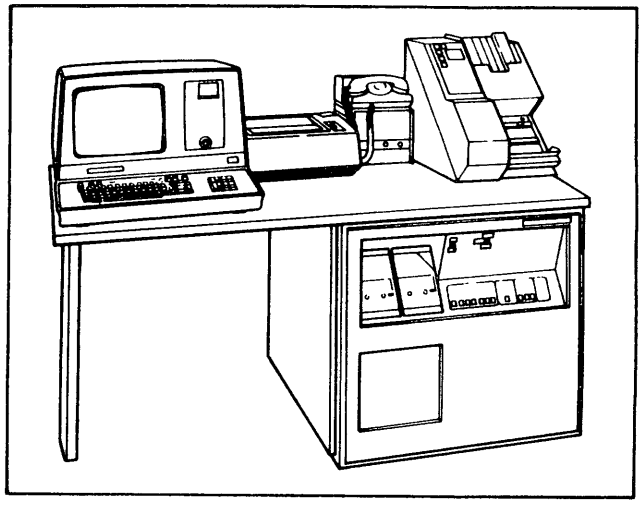


Figure 4-5. Remote Batch Terminal

5. What key ends a line of data? What key sends data? (Does the terminal use block mode?)

You should know when data leaves your terminal so that you know when to expect a response from the network or the host computer. Your terminal operator's guide should tell you what key or keys send data from your terminal.

Section 2 describes the keys and modes of terminal operation.

6. What telephone number is appropriate?

Your network might use a specific telephone number for terminals that communicate at a certain speed, for terminals operating in block mode, or for all terminals using certain protocols.

The telephone number you use determines whether you need to complete some of the procedures. If you do not know what numbers you should use, get help from a site administrator.

7. Is your terminal automatically connected to a host?

If it is, answer questions 8 through 10 below, then skip the subsection called Connecting Your Terminal to a Host.

8. What is the name or node number for the host you need to use?

9. Will your terminal be automatically logged in?

If so, answer question 10. Skip the subsection called Identifying Your Terminal to the Host.

10. What is the family name, user name, and password appropriate for you on the host system you will be using?

11. Is your terminal automatically connected to a network application program?

If so, skip the subsection called Connecting to a Network Application.

SETTING UP YOUR TERMINAL

This is step 2, as shown in figure 4-6.

If your terminal is hardwired, you can probably skip most of this subsection; hardwired terminals are usually set up once and left in the appropriate condition. Before using this subsection, review the applicable terminal operator's manual, which describes the procedures for setting up your terminal. If you are using an X.25 terminal, follow the PAD access procedures supplied by the packet-switching network.

Switch settings are important. The number, type, settings, and names of switches vary from terminal to terminal. Some terminals use buttons or toggles for switches; others use software option selections as switches.

If other people have used your terminal to connect to the network, the switches should be set correctly; you do not need to change them. Directions for setting the switches may be posted near your terminal.

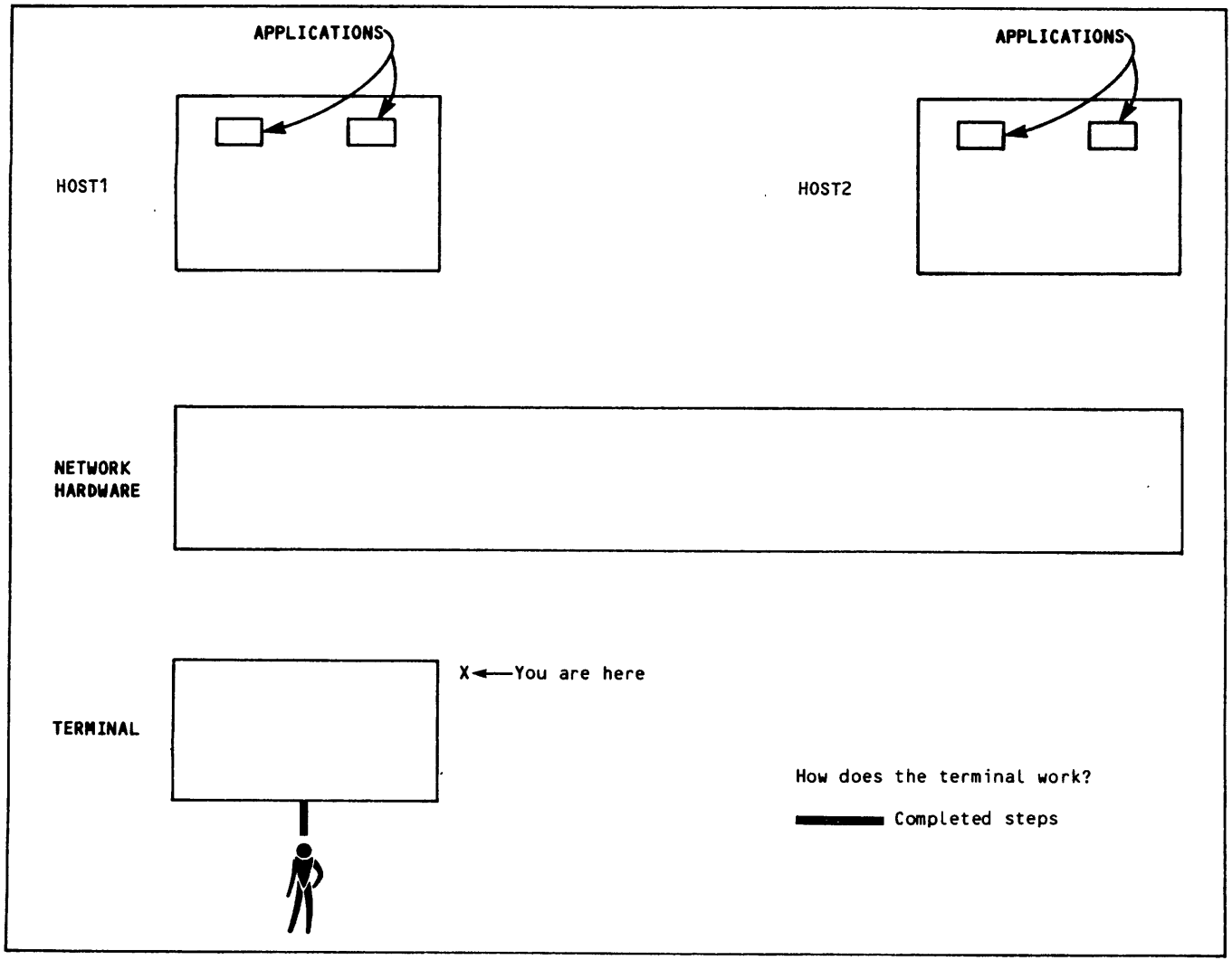


Figure 4-6. Step 2, Setting Up Your Terminal

Here is a suggested procedure to follow when setting up your terminal:

1. Turn on the power switch.

2. Load and initialize any software or controlware needed by your terminal. You will find detailed information on this task in your terminal's operator manual.

3. Set the terminal's duplex or echoplex switch to the correct position (use HALF if you are not sure). If nothing you enter appears on your console after you connect to the network, the network must display characters for your console. Section 5 describes how to enable and disable character echoing by the network (see the EP command).

4. Set the terminal's parity switch to the proper position. If garbled output appears at your console after you connect to the network, the network must use a different parity choice for your terminal. Section 5 describes how to change parity use by the network (see the PA command).

5. Set the line speed switch to a speed that matches one associated with the telephone numbers you were given. If you do not know the line speed you should use, set the switch to any position for X.25, mode 4 synchronous, or bisynchronous terminals; set the switch to 30 characters per second (300 bits per second) for asynchronous terminals.

6. Set the transmission mode switch to either the character or line mode position (you can change it later if you have a block mode terminal). For example: on a CDC 751, set the switch to the CHAR position; on a CDC 200 User Terminal, set the switch to the LINE position.

7. Set the on-line/off-line switch to the position that permits on-line communications. For example: on a CDC 721, select the ON option of the LINE setting; on a CDC 713, turn off the LOCAL indicator switch; on a Teletype, set the LINE/OFF/LOCAL switch to the LINE position; on a CDC 200 User Terminal console, set the ATTENDED/UNATTENDED switch to the ATTENDED position.

You are now ready to connect your terminal to the network.

CONNECTING YOUR TERMINAL TO THE NETWORK

To begin communication, your terminal must be physically connected to the network. This is step 3, shown in figure 4-7. If your terminal is hardwired, skip the remainder of this section; this section applies only to dial-up terminals.

If you are connected to a Local Area Network (LAN), you might have to use local procedures to use dial out resources or you might be able to connect directly to the network via the LAN. This book does not describe such procedures. In either case, skip the remainder of this subsection.

A dial-up terminal uses either an acoustic coupler (asynchronous terminals only) or a data set (asynchronous, X.25, bisynchronous, or synchronous terminals) to link itself to the network.

An acoustic coupler can be built into the terminal. Figure 4-8 shows a terminal with a built-in acoustic coupler and gives directions to connect the terminal to the network.

An acoustic coupler can also be separate from the terminal. Figure 4-9 shows an acoustic coupler separate from a terminal and gives directions to connect the terminal to the network.

A data set can be built into the terminal. Figure 4-10 shows a terminal with a data set built in and gives directions to connect the terminal to the network.

A stand-alone data set can have either a switch labeled TALK and DATA or individual buttons labeled TALK and DATA. Figure 4-11 shows a data set with switches and a data set with buttons, and gives directions to connect the terminal to the network.

After you dial the phone number, you either get the tone mentioned in the figures or an operator answers. If an operator answers, ask for your terminal to be connected, wait for the tone, then proceed as described in the figures. If you get a busy signal, wait and redial or try another number.

If your terminal has an indicator (sometimes marked DSR, DATA SET READY, or SYSTEM ACTIVE), it lights to let you know that the terminal is connected to the network. Wait approximately two seconds after the light comes on before entering data.

You are now ready for the next procedure. You might need to do one or all of the following:

Identify your terminal to an X.25 network (this book cannot describe such a procedure)

Identify your terminal to the CDC network

Select a host

Connect to a selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure usually sends you a message that helps you decide which procedure comes next. If nothing appears at your terminal within a few seconds, you probably need to identify the terminal to the network. Turn to the subsection appropriate for the next action you must perform.

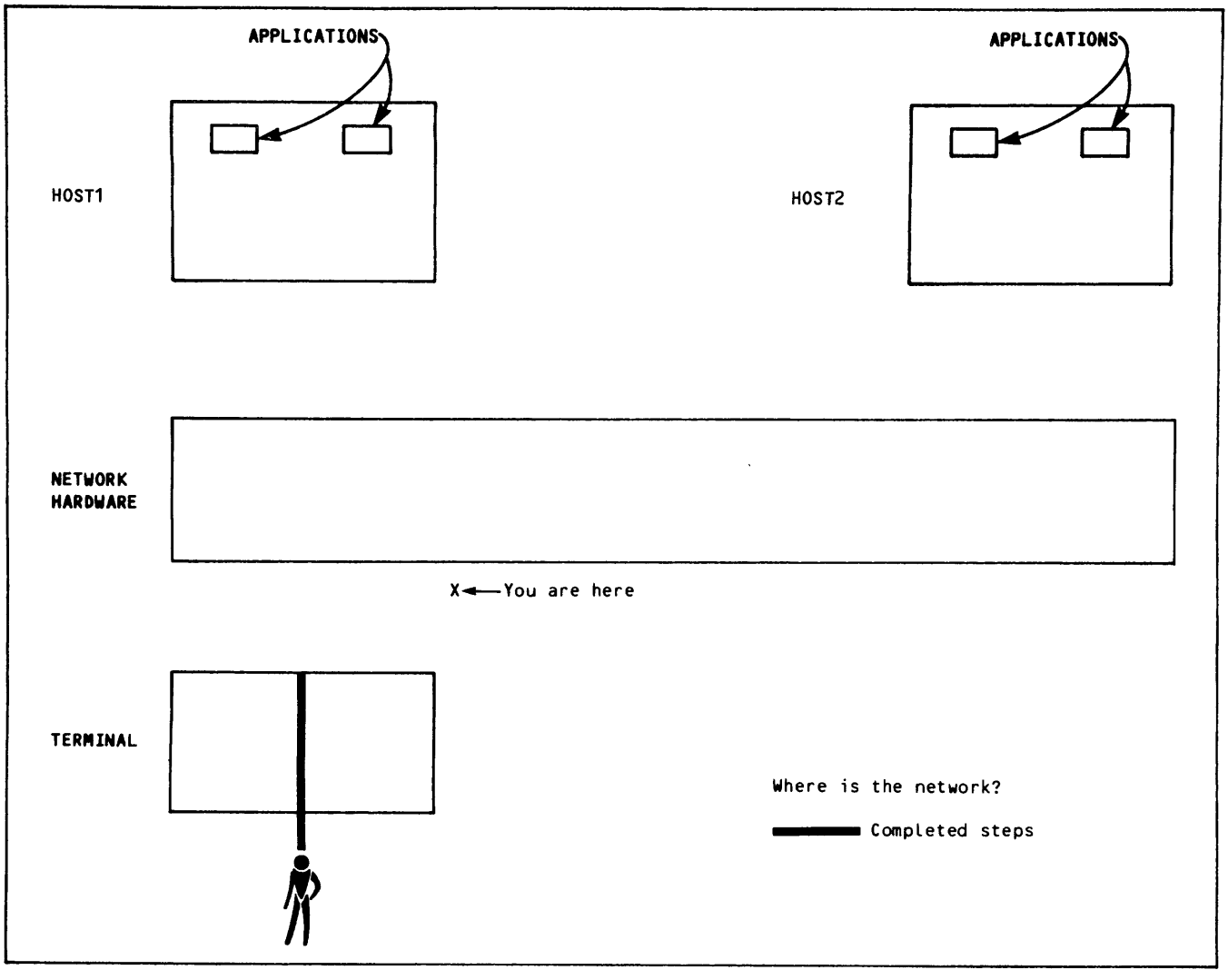


Figure 4-7. Step 3, Connecting to the Network

1. Pick up receiver.
2. Dial phone number.
3. Wait for high-pitched tone.
4. Fit receiver into coupler.

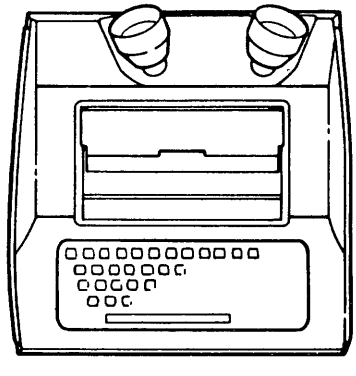


Figure 4-8. Terminal With Built-in Acoustic Coupler

1. Turn on coupler.
2. Pick up receiver.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Fit receiver into coupler.

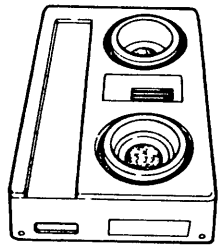


Figure 4-9. A Separate Acoustic Coupler

1. Pick up receiver.
2. Push button, such as ORIG button on Teletype model 33 or 35.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Push button, such as ORIG button on Teletype model 33 or 35.
6. Replace receiver.

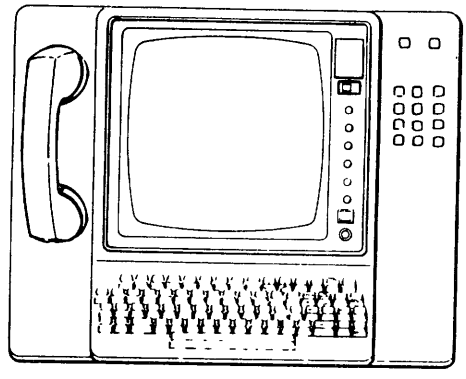
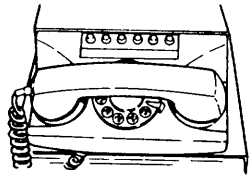


Figure 4-10. Data Set Built Into the Terminal

1. Pick up receiver.
2. Set switch to TALK or push TALK button.
3. Dial phone number.
4. Wait for high-pitched tone.
5. Set switch to DATA or push DATA button.
6. Replace receiver.

Data set with switch



Data set with buttons

Figure 4-11. Data Sets With Switches and Buttons

IDENTIFYING YOUR TERMINAL TO THE NETWORK

Once you have established physical connection to the network hardware, you might need to identify the terminal to the network software. This is step 4, shown in figure 4-12.

This identification procedure, called automatic recognition, is not always needed. It does not apply to X.25 or 3270 terminals. It does not apply if the communication line used by your terminal is not configured for automatic recognition.

If your terminal does not display any information within a few seconds of physical connection, you should complete one of the identification procedures described in the next subsections. The procedure you use depends on the terminal protocol, which you can determine if you know the terminal's type and/or class.

NOTE

If you are using a 2741-compatible terminal, you might have to press the carriage return key before output can appear at your terminal. If that action locks your device's keyboard, you can unlock it again by momentarily switching the terminal off-line (into LOCAL mode), then back on-line. Output can then appear on the device.

If you do not complete the identification procedure within the allowed time, the network disconnects a dial-up terminal. Hang up the phone and redial the number if you are using a dial-up terminal. You must restart the procedure for a hardwired terminal.

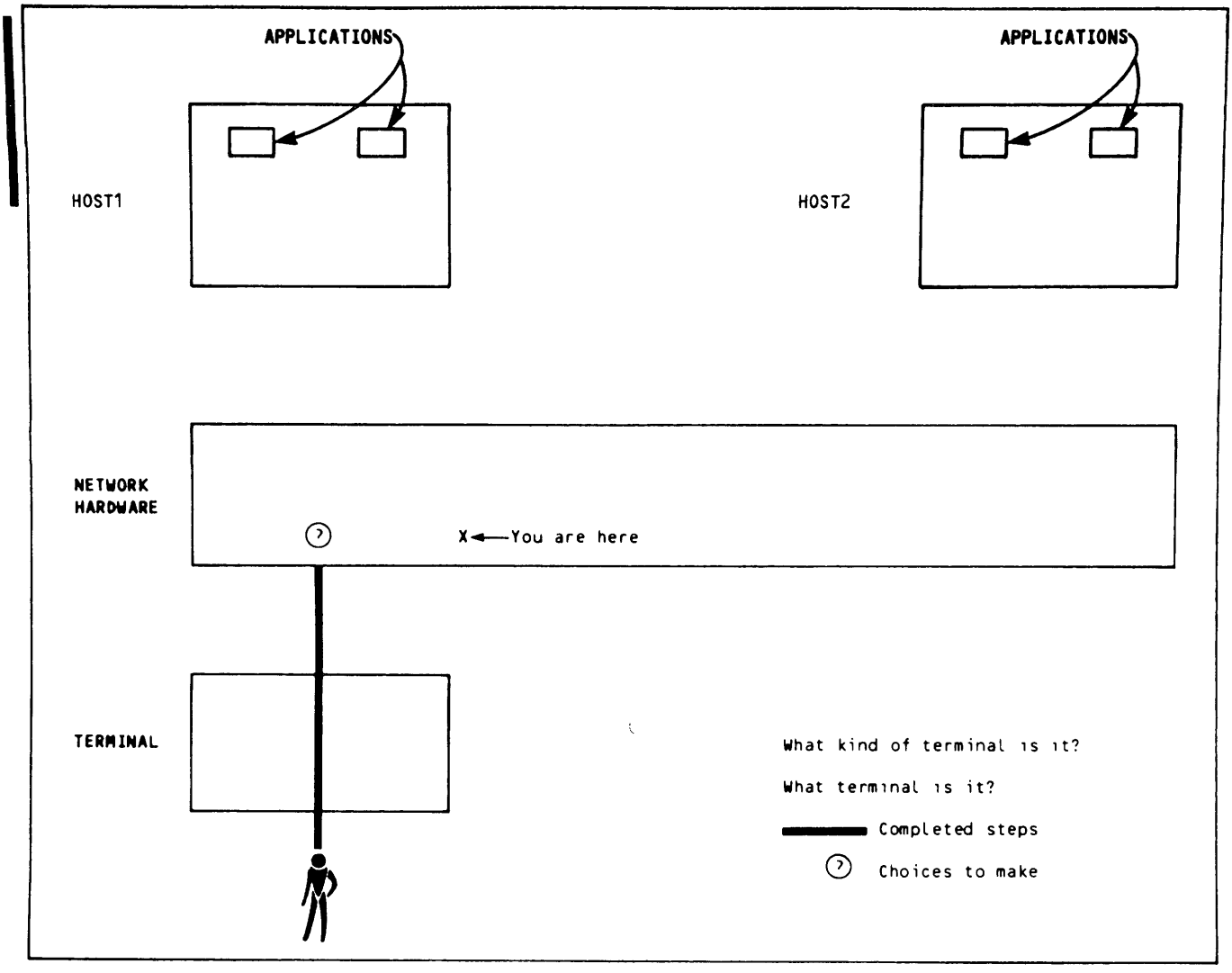


Figure 4-12. Step 4, Identifying Your Terminal to the Network

PROCEDURE FOR ASYNCHRONOUS TERMINALS

For an asynchronous terminal, which belongs in terminal classes 1 through 8, complete this procedure within 1 minute:

1. Wait 2 seconds after the light comes on indicating your terminal is connected to the network.
2. Press the carriage return key to identify the line speed used by your terminal. The network software responds with two line feeds.
3. If your terminal uses an APL, EBCD, or Correspondence code character set, type a right parenthesis; if your terminal uses an ASCII character set, you need not type in anything.

   Press the carriage return key to identify the character and code set used by the terminal. The network software responds with one line feed.

You are ready for the next procedure. You might need to do one or all of the following:

Select a host

Connect to a selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

PROCEDURE FOR HASP TERMINALS

HASP terminals belong either to terminal class 9 or 14. For automatic recognition of your HASP terminal, you might need to modify the signon block or enter a special statement.

The first information transmitted by a HASP workstation after it is connected to the network is called a signon block. A workstation might handle signon block transmission in one of several ways.

- Some workstations automatically transmit a signon block that begins with the characters

  /*SIGNON

  These eight characters and information following them imitate the card required by some IBM host systems for workstation identification. This transmission can occur without your knowledge or intervention if the information is built into the firmware or hardware.

- Some workstations require you to enter information for the signon block before they will transmit anything to the network; however, they only allow the block to contain information beginning with the eight /*SIGNON characters. These characters and other information might be required as:

  A command from the console

  A card read from any card reader

  A card read from a specific card reader

- Some workstations require you to enter signon block information before they will transmit anything to the network, but the content of the signon block is left up to you. Whether the information can be entered from the console or from a card reader depends on the workstation.

The network software cannot use information from a /*SIGNON card image. When such a card image is received in a signon block, the signon block contents are discarded.

Instead, the network software uses a /*CONFIG statement for the automatic recognition procedure. You can send the /*CONFIG statement in the signon block when your terminal permits you to provide the contents of that block.

If the signon block does not contain the /*CONFIG statement and the terminal must use automatic recognition, a prompt requesting that statement is sent to the workstation console. When a prompt for the statement appears, enter the statement through a card reader.

The format of the /*CONFIG statement is:

/*CONFIG[,ti,CO=co,CR=x,LP=x,CP=y,PL=z]

Blanks are not allowed within the statement; a blank ends parameter processing. The parameters, which you can specify in any order, are as follows:

ti Is the terminal type indicator:

- POST HASP postprint (default); this is terminal class 9
- PRE HASP preprint; this is terminal class 14

co Is the configuration ordinal, a decimal integer from 1 to 255; default is 1. Use the configuration ordinal to select one of several terminal definitions defined for the line that describes a specific combination of additional characteristics. Site administration personnel can tell you the correct number to use.

CR=x Identifies available card readers. If omitted, there is no such device.

LP=x Identifies available line printers. If omitted, there is no such device.

CP=y Identifies available card punches; y cannot equal z. If omitted, there is no such device.

PL=z Identifies available plotters; y cannot equal z. No default exists for z; you cannot omit PL=z if you have a plotter.

where x, y, or z is a list of numbers specified by

1/2/.../7

or

ALL All numbers from 1 to 7

The numbers you use for x, y, and z must match the HASP stream numbers used for the corresponding devices within the workstation. Site administration personnel should have this information if you do not know what numbers to use.

For example:

/*CONFIG,PRE,CO=2,CR=2,LP=2/3,CP=4,PL=5

identifies your terminal as the second preprinting workstation defined for the communication line, having:

A card reader and line printer on stream 2

Another printer on stream 3

A card punch on stream 4

A plotter on stream 5

If no prompt for a /*CONFIG statement appears, you are ready for the next procedure. You might need to do one or all of the following:

Select a host

Connect to a selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

PROCEDURE FOR MODE 4 TERMINALS

Mode 4 terminals belong to terminal classes 10 through 13 and 15. For automatic recognition, press the end-of-block key within one minute of physical connection.

You are ready for the next procedure. You might need to do one or all of the following:

Select a host

Connect to a selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

PROCEDURE FOR BISYNCHRONOUS TERMINALS

Bisynchronous terminals belong to terminal classes 16 and 17. Enter a /*CONFIG statement to specify the type of terminal and the devices available at it.

The format of the /*CONFIG statement is:

```
/*CONFIG[,ti,CO=co,CR,LP,CP[=y]]
```

Blanks are not allowed within the statement; a blank ends parameter processing. The parameters, which you can specify in any order, are as follows:

ti — Is the terminal type indicator:

- 2780 — IBM 2780 (default); this is terminal class 16
- 3780 — IBM 3780; this is terminal class 17

co — Is the configuration ordinal, a decimal integer from 1 to 255; default is 1. Use the configuation ordinal to select one of several terminal definitions defined for the line that describes a specific combination of additional characteristics. Site personnel can tell you the correct number to use.

CR — Indicates an available card reader; assumed if not specified.

LP — Indicates an available line printer; assumed if not specified.

CP=y — Indicates an available card punch; assumed not to exist if not specified. The =y portion is not allowed for 2780; for 3780, the y portion is required and can be either 2 or 3 to correspond to the device selection character DC2 or DC3.

For example:

```
/*CONFIG,3780,CO=4,cr,LP,CP=2
```

identifes your terminal as the fourth 3780 configuration defined for the communication line, having:

A card reader and line printer

A card punch which is selected by the character code DC2

If no prompt for a /*CONFIG statement appears, you are ready for the next procedure. You might need to do one or all of the following:

Select a host

Connect to a selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure sends you a message which helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

PROCEDURE FOR IBM 3270 TERMINALS

IBM 3270 terminals belong to terminal class 18. Automatic recognition is not performed for these terminals.

You might need to do one or all of the following:

Select a host

Connect to the selected host

Identify yourself to the selected host (log in)

Select a network application

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

SELECTING A HOST

You might need to select the host that runs the software you want to use. This is not necessary if your installation personnel select a path to a host for your terminal when the network is configured.

You can select or change a site-defined host path after you connect your terminal to the network. The host path selected is used either until you change it or until your terminal is disconnected from the network.

More than one host path can be available to each host from a terminal. If one host path fails, you can select an alternate route. Selecting your host path is step 5, as shown in figure 4-13.

SELECTING ANY HOST PATH

You can select a host path or override a site-defined host path selection with the following terminal definition command:

```
ct HS=hostname [cr]
```

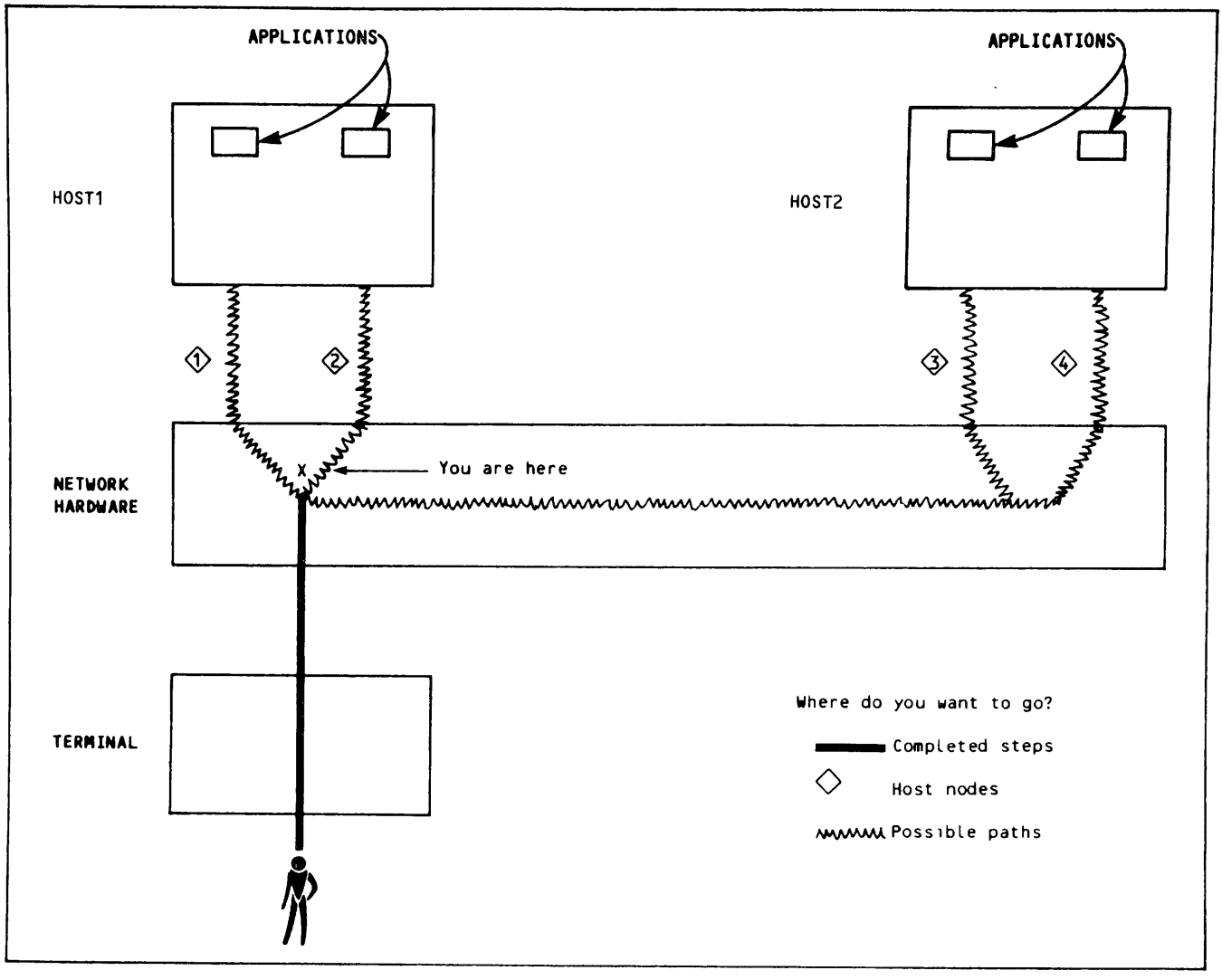


Figure 4-13. Step 5, Selecting a Host Path

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

Each host has a unique name. This name (one through seven characters) is indicated by hostname. Your terminal uses the path to that host with the least traffic.

For example:

```
%HS=SVLNOS  cr
```

would select the path with the least traffic to the host called SVLNOS (% represents your terminal's network control character).

If you do not know the name you need, the host availability display (HAD) can show you what names are available. That display is described later in this section.

Once you have selected your host, you are ready for the next procedure. You might need to do one or all of the following:

Connect to the selected host

Identify yourself to the selected host (log in)

Select a network application program

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

SELECTING A SPECIFIC HOST PATH

Use the following terminal definition command to select your own host path or to override the site-defined host path:

```
ct HN=nn  cr
```

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

Each host path has a unique number, called a node number. The host node number (1 through 255) is indicated by nn.

For example:

```
%HN=2   cr
```

would select host path 2 (% represents your terminal's network control character).

If you do not know the node number you need, the host availability display (HAD) can show you what numbers are available. That display is described in the following subsection. Each node number has a name for its host, provided by your site to help you identify the corresponding host system on the HAD.

Once you have selected the host path, you are ready for the next procedure. You might need to do one or all of the following:

Connect to the selected host

Identify yourself to the selected host (log in)

Select a network application program

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

CONTROLLING THE DISPLAY OF HOST PATHS

As shown in figure 4-14, the host availability display lists all host paths your terminal can currently use to access any host in the network. You can request the display with the following terminal definition command:

```
ct HD [cr]
```

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

CONNECTING TO THE SELECTED HOST

Your site personnel might have configured your terminal so that it is automatically connected with a host. If so, the network continues to attempt connection indefinitely until the connection is completed. If not, you will have to ask for the connection to be made. This is step 6, as shown in figure 4-15.

You can detect the network's repeated attempts at connection by requesting the host availability display. The connection attempts show as an SA status for one of the hosts.

You can stop the network's repeated connection attempts using the TM command, described later in this section.

You can ask to be connected to the selected host by entering a line. If the entry is not an empty line, the entry is discarded (unless it is a valid terminal definition command).

If your connection is rejected, you are notified. If you select another host path, you must enter another type-in to complete a connection. The network software physically disconnects dial-up devices if you do not attempt connection to a host within two minutes of completing the previous procedure.

HOST CONNECTION COMMAND

The host connection (HC) command allows you to terminate the connection between your terminal and a host or override a site-defined host path selection. Its format is:

```
ct HC [=hostname] [cr]
```

where hostname is the name of the desired host. The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

The shorter version of the HC command terminates the current connection and then establishes a new connection to the currently selected host, which might be different from the currently connected host.

The longer version of the HC command ends the connection to the current host, selects a new host, and establishes the connection to the new host.

For example:

```
%HC=SVLNOS [cr]
```

terminates the connection between your terminal and a host and selects the host path with the fewest connections to the host called SVLNOS. The % character represents the network control character for your terminal.

Once you are connected to a host, you are ready for the next procedure. You might need to do one or both of the following:

Identify yourself to the host (log in)

Select a network application program

The software expecting the next procedure sends you a message that helps you decide which procedure comes next. Turn to the subsection appropriate for the next action you must perform.

IDENTIFYING YOURSELF AND YOUR TERMINAL TO THE HOST

The procedure that identifies you and your terminal to the host system is called login. This is step 7, as shown in figure 4-16. Your terminal is identified to the host system by assigning it a family name, a user name, and a password. The host can prompt you for each of these entries.

Your site can set up your terminal connection so that part or all of your login is done automatically. If so, you might not see the corresponding prompt. If not, you must complete some or all of the login procedure yourself.

```
host status message              CONTROL CHARACTER = (ct)
NPU NODE = nn                    TERMINAL NAME = tttt
HOST            NODE             SELECTED/          STATUS
                                 CONNECTED
host1           node1            condition1         status1
host2           node2            condition2         status2
  .               .                  .                .
  .               .                  .                .
  .               .                  .                .
hostn           noden            conditionn         statusn
prompt message
```

LEGEND:

informative message — Status of a given host, which can be one of the following:

		You also see:
HOST CONNECTED	You are connected to the host	prompt message d or e
NO HOST SELECTED	Host is available; you have not selected a host	prompt message b
NO HOST AVAILABLE	No hosts are available	prompt message a
HOST UNAVAILABLE	You have selected a host that is not available	prompt message a
HOST BUSY	Connection rejected by host	prompt message a
HOST DISCONNECTED	Connection to host terminated	prompt message c or e
HOST AVAILABLE	You have selected a host that is available	prompt message c
NO HOST CONNECTED	You are not connected to a host	prompt message c
RECONFIGURING	You are connected to a host	prompt message c

ct — Network control character currently defined for your terminal.

nn — The NPU node number currently associated with your terminal.

tttt — The name associated with your terminal.

hosti — The 1- to 7-character name of a host; if the network only has one host, this can be blank.

nodei — The host node number used in selecting a host path through the network; $1 \leq nodei \leq 255$.

conditioni
- S = selected, not connected
- C = connected, not selected
- S C = selected and connected
- S A = selected, attempting connection

statusi — Either AVAILABLE or NOT AVAILABLE; a host is only available when it is connected to the network.

prompt message — Action to take in response to a host status message, which can be one of the following:

- prompt message a: ENTER ct HD TO SEE HOST STATUS
- prompt message b: ENTER ct HN=nn or ct HS=hosti TO SELECT HOST
- prompt message c: ENTER INPUT TO CONNECT TO HOST
- prompt message d: READY FOR INPUT
- prompt message e: TERMINAL DISABLED BY NOP

Figure 4-14. Host Availability Display

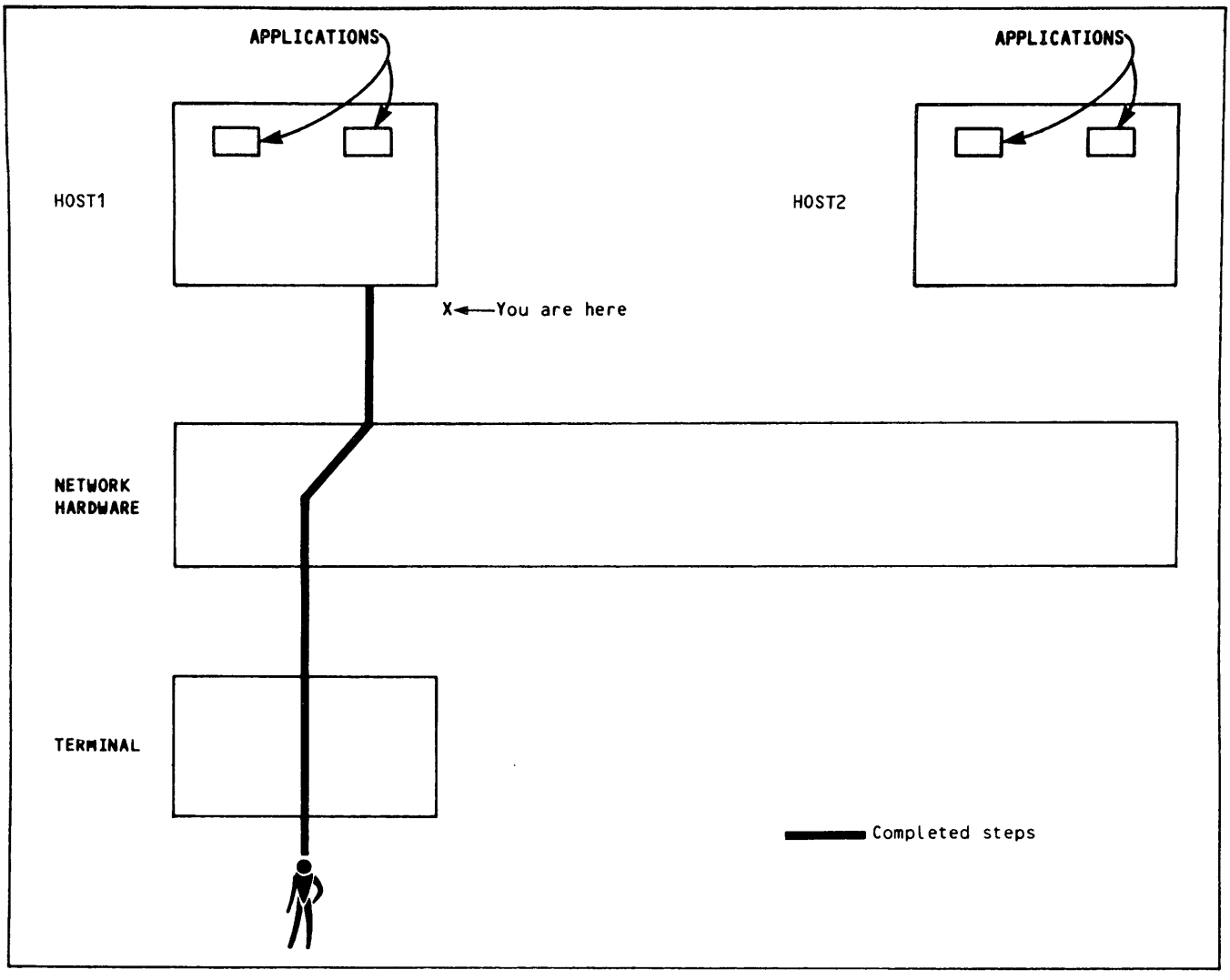


Figure 4-15. Step 6, Connecting to a Host

Your site also might use a host access secure login character to ensure that you send your login information to the correct host software. The character used is site-defined. If you have been told to use such a character, enter it now in the following way:

At an asynchronous terminal other than a 2741, press the interactive interrupt key (BREAK or ATTN), and type the character.

At a 2741 terminal, press the NEW LINE key twice, the ATTN key, type the character, and press [cr].

At an X.25 terminal, press the interactive interrupt key, type the character, and press the PAD forwarding signal key.

At a mode 4 terminal, clear the screen, press [cr], clear the screen again, type the character and press [cr].

At a HASP terminal, type the character and press the end-of-line key.

At a 2780 or 3780, enter /* and the character from the console or a card reader.

At a 3270, clear the screen, press the program attention 1 key, type the security character, and press the ENTER key.

This action momentarily disconnects you from the host. You might need to reconnect your terminal to the host (if connection is not automatic). Now you are guaranteed that you are sending your login entries to the correct host software.

Figure 4-17 shows a sample login from an asynchronous terminal. In the example, the terminal has not been configured by site personnel for automatic login. Everything you would typically enter is shown in lowercase.

AUTOMATIC LOGIN

The family name or user name assigned to your terminal for an automatic login can be:

A mandatory value

A default value

A primary value

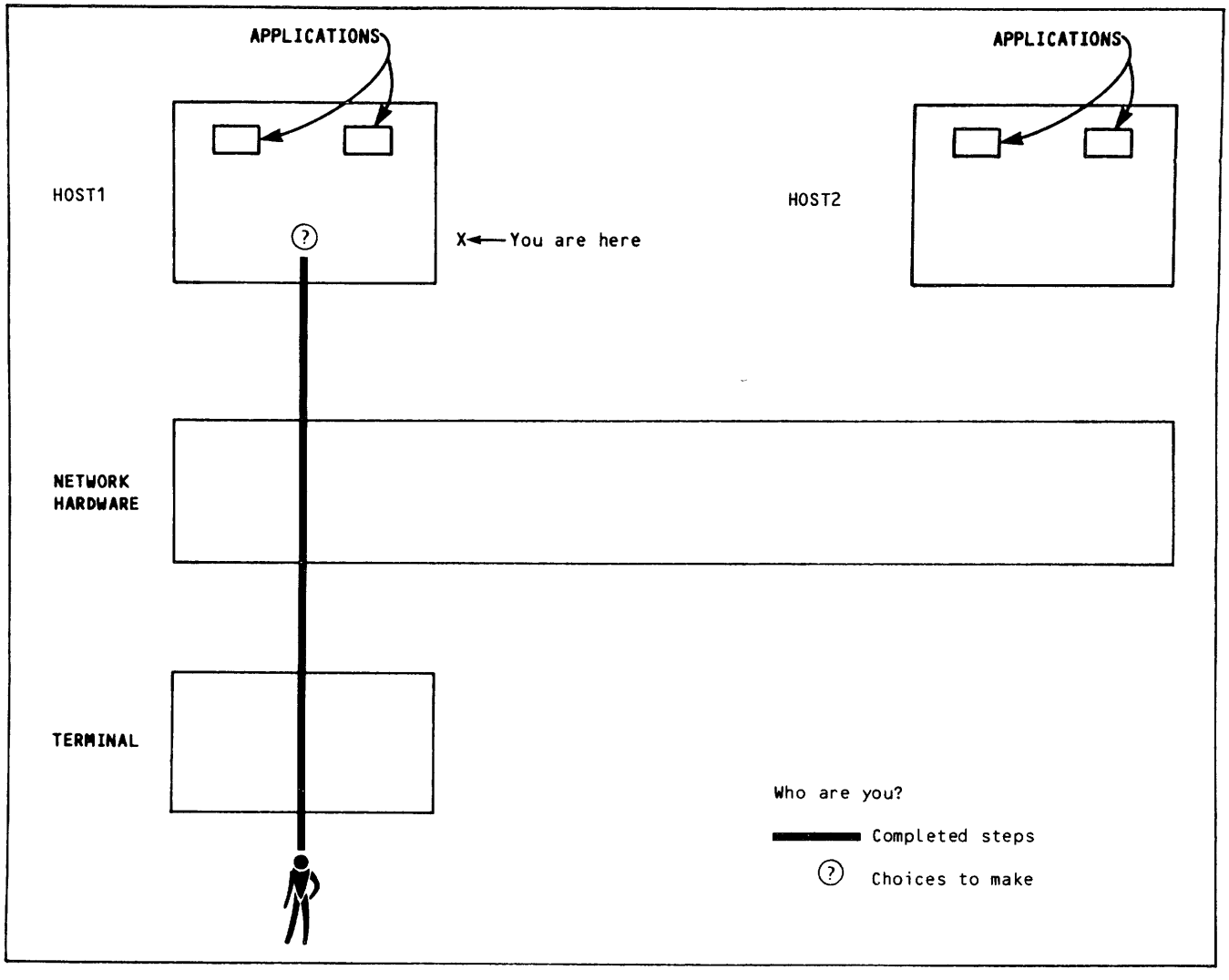


Figure 4-16. Step 7, Identifying Your Terminal to the Host

Terminal Dialog	Comments
WELCOME TO THE NOS SOFTWARE SYSTEM. COPYRIGHT CONTROL DATA 1978, 198x.	System identifier.
yy/mm/dd. hh.mm.ss. TERM201	Date, time, and terminal name.
CDC Network Operating System NOS 2	Identifying installation header.
FAMILY: systema [CR]	Response to system prompts.
USER NAME: rls4525 [CR]	
PASSWORD:	
XXXXXXX [CR]	Type your password over the blacked-out characters.

Figure 4-17. Sample Login

If a mandatory value is assigned, you do not receive a prompt and cannot use any other value.

If a default value is assigned, you are prompted, and you can either: enter an empty line as a response to the system prompt (if you want either the default family or user name to apply) or respond to the prompt with another value.

The default family name value assigned to your console for automatic login can be different from the default value used by the host for a family name. You can override the login default even when you do not know the system default; this procedure is described in the following subsection.

If a primary value is assigned, you do not receive a prompt the first time login occurs while you are connected to the host. Subsequent logins (see Restarting Host Identification) do prompt you. You can override a primary value for user name by entering a different value during an abbreviated login. Abbreviated login is described in a later subsection.

LOGIN DIALOG

Login begins when the system displays the following lines:

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198x.
```

Then a line appears with the date, time, and terminal name in this format:

```
yy/mm/dd.   hh.mm.ss.   TERM201
```

Next is a line your site supplies that identifies the host system used; something similar to:

```
CDC SUNNYVALE OPERATIONS  NOS 2 PSR LEVEL xxx
```

The next line is either of the following prompts:

FAMILY:

This indicates that no mandatory or primary family name has been assigned to your terminal. Enter the one- through seven-character name of the storage device that contains your permanent files and press the carriage return key. If you want to use the default login family name, just press the carriage return key after the FAMILY prompt appears. You can also enter a value of 0 (zero) to override a preassigned login default family name and use the system default family name.

USER NAME:

This indicates that no mandatory or primary user name has been assigned to your terminal. Enter the user name you were given. The user name, which can contain any combination of digits, letters, and asterisks, identifies you as the terminal operator.

The next prompt appears only after a USER NAME prompt has appeared. This final prompt is:

PASSWORD:

or

PASSWORD:

The system attempts to preserve your password's secrecy. If the network is echoing all input back to the terminal and you get the first form of the PASSWORD prompt, the network does not echo characters while you are entering your password. You will not see what you type in. At some terminals, the network overprints several characters on a line and asks you to type your password on that line, as shown by the second PASSWORD prompt. After creating this row of overstruck characters, the cursor moves back to the first character.

Enter the password currently associated with your user name. If you must enter the user name, you also must enter the password. No default exists for a password.

In addition, some sites might require some users to enter a personal id.

You must respond to each prompt within two minutes. If you take too long to respond, a timeout message is displayed at your terminal. Your terminal is then disconnected from the host.

You must spell each login entry correctly. If you make a mistake, such as typing the letter O for the numeral 0 or the number 1 for the letter l, your login will not be successful. Try again when the prompting sequence restarts.

You are allowed four consecutive chances to complete the login procedure. If you make four unsuccessful attempts to complete login, a message appears at your terminal. Your terminal is then disconnected from the host. If you are using a dial-up line, the line is then disconnected. If this happens, check the accuracy of all of the entries. If you entered everything correctly, contact a site administrator.

When the family name, user name, and password are accepted by the host, you are ready for the last procedure: connecting your terminal to a network application program.

CONNECTING TO A NETWORK APPLICATION

This is step 8, shown in figure 4-18. Your site can set up your terminal connection so that this step is done automatically. If so, you might not see the corresponding prompt. If not, you must complete this procedure yourself.

You are allowed several chances to make a correct entry. However, if you fail to connect successfully after four consecutive attempts, your terminal is automatically disconnected from the host. If this happens, check the spelling of the name you used. If the spelling was correct, contact a site administrator.

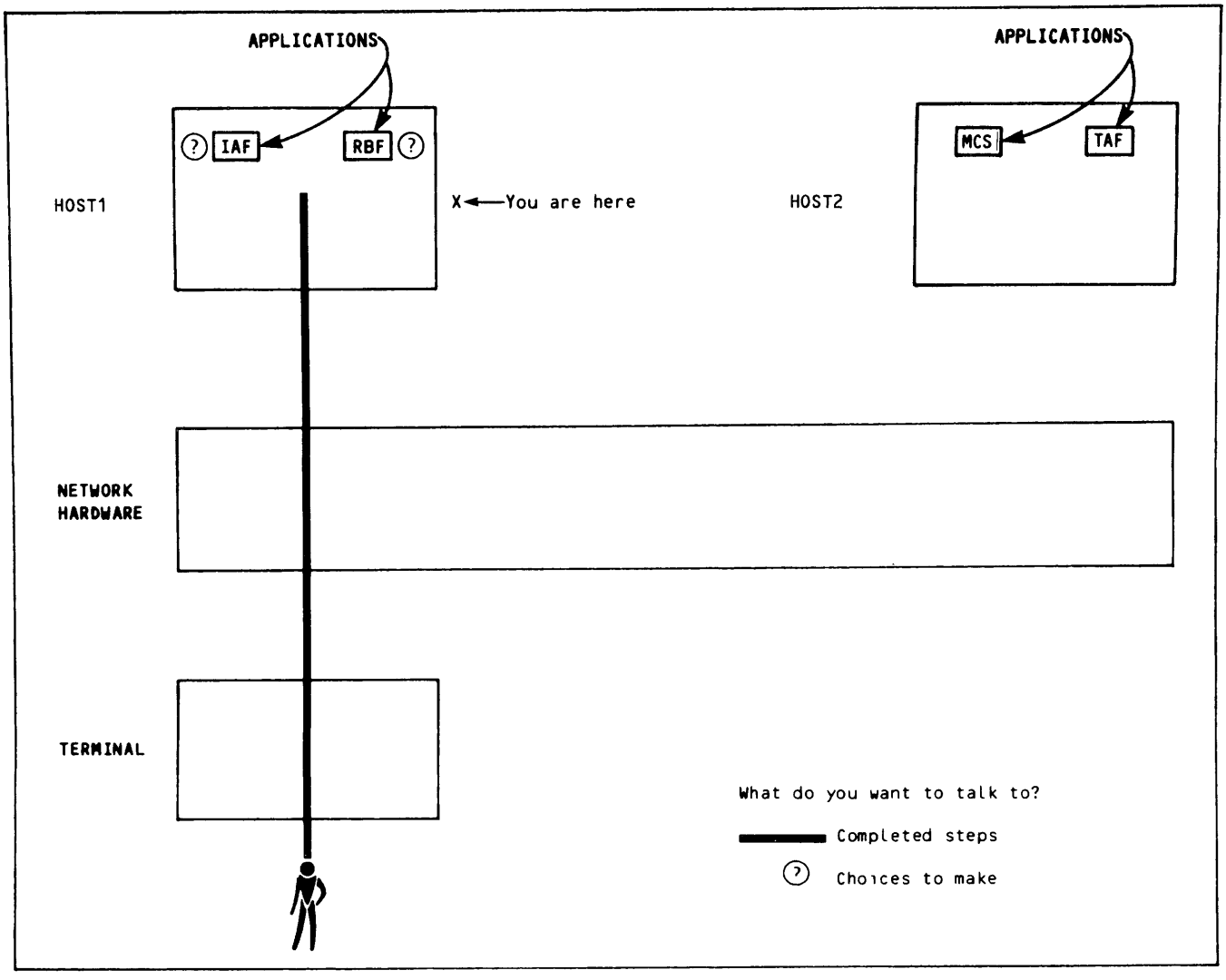


Figure 4-18. Step 8, Connecting to a Network Application

AUTOMATIC CONNECTION

Your host may permit you access only to one network application program, or an application program can be connected automatically. Your host site can configure your terminal for either possibility by presassigning an automatic connection application name.

A preassigned automatic connection application can be either mandatory or primary for the terminal or it can be mandatory for the user. If either you or your terminal has a preassigned mandatory application, you are not prompted, you are automatically connected to that application, and you cannot use any other application. If a mandatory application program is not running in the host, you are logged out and disconnected from that host.

If a primary value is assigned, you are not prompted for the initial connection request after identifying your terminal to the host, and connection to that program occurs automatically. If that connection is not successful or if you disconnect from that application program, you have a choice of actions; you can:

Select another application

Disconnect from the host

To override a primary automatic connection application on the first attempt to select an application, name a different application in an abbreviated login entry (described later in this section) following a login prompt.

When a primary value is assigned, you are prompted on subsequent attempts to select an application, such as when you are switching applications. When you are prompted to enter an application name and you want to be connected to your primary application, enter an empty line as a response.

MANUAL CONNECTION

This procedure begins when you see the following prompting message:

terminalname - APPLICATION:

The terminal name variable on this line is the same terminal name as the one on the first line of your login sequence. If your terminal is automatically logged into a network application program, you do not receive this prompt.

When you receive the APPLICATION prompt, enter the letters and digits that identify the network application program you want to access; for example, IAF for the Interactive Facility. You can access CDC-written application programs from the following list that are installed at your site:

Remote Batch Facility (enter RBF)

Interactive Facility (enter IAF)

Transaction Facility (enter TAF)

Terminal Verification Facility (enter TVF)

Communications Supervisor (enter CS)

Message Control System (enter MCS)

PLATO NAM Interface (enter PNI)

NOS/VE Interactive Facility (enter VEIAF)

Interactive Transfer Facility (enter ITF)

Other application programs, written at your site, might be available.

You must respond to the prompt within two minutes. If you take too long to respond, a timeout message is displayed at your terminal. Your terminal is then disconnected from the host.

You must spell your entry correctly. If you make a mistake, such as typing the letter O for the numeral 0 or the letter l for the number 1, you receive an error message.

If you make four unsuccessful attempts to enter an application name, a message appears at your terminal. Your terminal is then disconnected from the host. If you are using a dial-up line, the line is then disconnected.

This procedure is complete when the application you request connects to your terminal (see figure 4-19). The application you asked to use might print an identification line next.

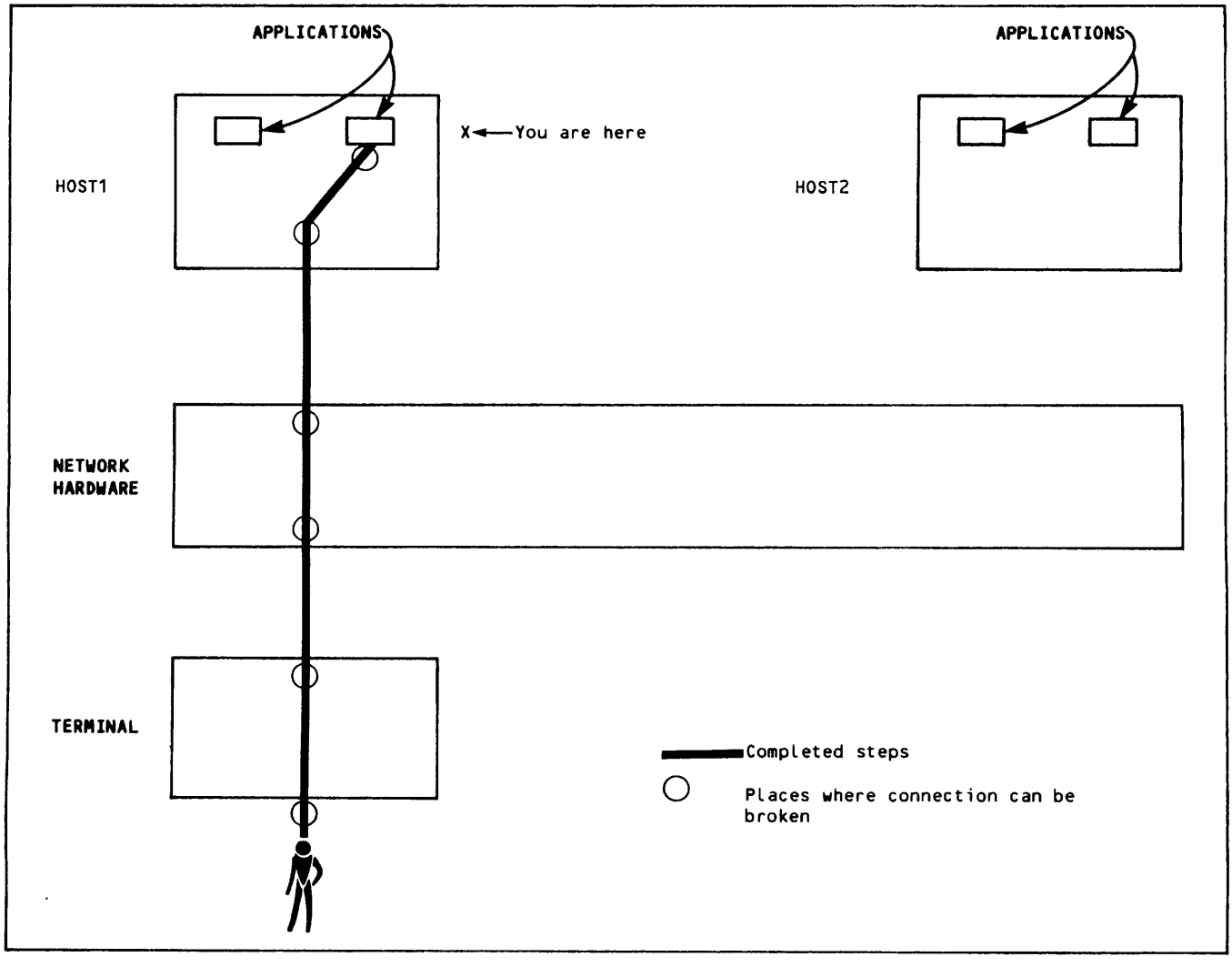


Figure 4-19. Step 9, Connection Completed

Whether the application sends such a message or not, you can leave this section of the book and use commands recognized by that program. If you need to disconnect your terminal from the application, the application probably has a command for that purpose. If the application has no such command, the subsection called Disconnecting From the Host describes a command you can use.

ABBREVIATED LOGIN AND APPLICATION CONNECTION

You can shorten the standard login and application connection procedures by entering all of the required information at once, unless you are required to enter a personal id. For example, in response to the FAMILY prompt, you can enter the following information on the same line:

```
FAMILY:
familyname,username,password,application  cr
```

Separate all entries with a comma. The entries are order-dependent. Use a comma to indicate any default or unused entries.

You do not receive any of the remaining login or application selection prompts if you use this abbreviated login procedure, unless you omit a required entry or supply an entry that is not recognized. Your password entry is not protected by the system when you use an abbreviated login procedure.

Here are some examples. To log in and connect to the Remote Batch Facility application program RBF with a family name of systema , a user name of rls4525 , and a password of pass123 , enter:

```
systema,rls4525,pass123,rbf  cr
```

To use the site-defined login default family name, enter:

```
,rls4525,pass123,rbf  cr
```

To override your site-defined login default family name with the host's default family name, enter:

```
0,rls4525,pass123,rbf  cr
```

To use a login default family name and user name defined by the site for your terminal, enter:

```
,,,rbf  cr
```

Note that passwords are not required when a default user name is used.

To also use a default application name defined by the site for your terminal, enter:

```
,,,  cr
```

You can supply combined entries in response to the FAMILY, USER NAME, and PASSWORD prompts. Although you do not need to complete the remaining sequence on the same line, you must enter the information in the correct order: family name, user name, password, and application. For example:

```
FAMILY: systema  cr
USER NAME: rls4525,pass123  cr
terminalname - APPLICATION: iaf  cr
```

If you supply a user name/password combination or an application name that is overridden by a mandatory specification, a message is displayed and your supplied value is ignored.

SWITCHING TO A DIFFERENT APPLICATION

Unless you have a mandatory network application program assigned for your terminal, you can transfer your terminal's connection from the current application program to another application. There are two ways to do this, depending on the commands recognized by the current application program.

DIRECT SWITCHES

You can switch in one step by entering a switch command recognized by the current application. The following switch commands are recognized by CDC-written network application programs:

Program	Command
CS	none
RBF	IAF cr
IAF	BYE,application cr or APPSW,application,op cr
MCS	LOGIN,application cr
TVF	none
TAF	none
PNI	none
VEIAF	none

For example, if your terminal is connected to RBF, you can gain access to IAF by entering the IAF command.

Do not enter any more input until the application acknowledges that it has processed your switch command. Otherwise, you may lose typed-ahead input.

Using the APPSW,application,op command allows you to switch from IAF to a second application program without losing your IAF environment. You can, for example, use this command to switch from IAF to RBF. Then, when you type END to terminate your connection to RBF, you are automatically reconnected to IAF and can proceed where you left off with interactive usage. You don't need to save your files when you exit from IAF; and, when you reenter IAF you don't need to GET/ATTACH your files or select your subsystem.

In the future, other application programs, such as PLATO (PNI), might support indirect application switches without loss of your environment. Refer to your PLATO user documentation for details.

INDIRECT SWITCHES

If your current application program does not have a command for a direct switch, you must disconnect your terminal from the current program with a command recognized by that application. Then you must request another application connection from the host. The following disconnection commands are recognized by CDC-written network application programs:

Program	Command
CS	none
RBF	END [cr]
IAF	BYE [cr]
MCS	END [cr]
TVF	END [cr]
TAF	EX.LOGT [cr]
PNI	none
VEIAF	LOGOUT [cr]

For example, you can switch your console connection from RBF to an application program other than IAF by entering the END command. You then receive a prompt for a new application name. Do not enter any more input until the application acknowledges that it has processed your disconnection command. Otherwise, you might lose typed-ahead input.

DIALOG FOR DIRECT AND INDIRECT SWITCHES

After disconnecting the application program, the system indicates that a connection switch is in progress by issuing this message:

```
application - CONNECT TIME hh.mm.ss.
```

The application program that was just disconnected is indicated by application, and the time that passed while the terminal was connected to the program is indicated by hh.mm.ss.

If you used a switch command, you receive an identifying message from your new application program. If you used a disconnection command, you receive another prompt for an application program name:

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application program. If so, you are disconnected from the host.

If you enter the name of another application program in response to the APPLICATION prompt, you are connected to that application program.

RESTARTING HOST IDENTIFICATION

There are many reasons why you might want to restart the host identification procedure and the subsequent application connection procedure. Perhaps you have several user names available to you, each with permission to use different host resources. Or perhaps someone else left the terminal connected to an application you do not need, and you do not want to use their user name.

There are two ways to restart login, depending on the commands recognized by the current application program.

DIRECT RESTARTS

You can restart the host identification procedure in one step by entering a login command recognized by the current application. The following login commands are recognized by CDC-written network application programs:

Program	Command
CS	none
RBF	LOGIN [cr]
IAF	HELLO [cr]
MCS	LOGIN [cr]
TVF	none
TAF	none
PNI	none
VEIAF	none

For example, if your terminal is connected to RBF, you can restart login by entering the LOGIN command.

Do not enter any more input until the application acknowledges that it has processed your command. Otherwise, you may lose typed-ahead input.

INDIRECT RESTARTS

If your current application program does not have a command for a direct switch, you must disconnect your terminal from the current program with a command recognized by that application. Then you must request restart of login from the host. The disconnection commands recognized by CDC-written network application programs are described earlier in this section under Switching to a Different Application.

For example, you can restart your console login from RBF by entering the END command. You then receive a prompt for a new application name. Do not enter any more input until the application acknowledges that it has processed your disconnection command. Otherwise, you might lose typed-ahead input.

After entering the disconnection command, you receive another prompt for an application program name:

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application program. If so, you are disconnected from the host and can restart login by reconnecting.

If you receive the APPLICATION prompt, you use one of the following commands to ask the host system to restart login:

HELLO [cr]

LOGIN [cr]

DISCONNECTING FROM A HOST SYSTEM

You might want to disconnect your terminal from a host system for several reasons:

You want to disconnect from the network properly, without leaving behind an indication that something went wrong; disconnecting from a host is the first step in disconnecting from a CDC network.

You are allowed access to more than one host system, and you want to transfer your terminal's connection from the current host to another host.

There are three ways to do this, depending on the commands used by the current application program.

DISCONNECTING WITH APPLICATION PROGRAM COMMANDS

You can disconnect in one step by entering a host disconnection command recognized by the current application. The following host disconnection commands are recognized by CDC-written network application programs:

Program	Command
CS	BYE or LOGOUT or GOODBYE cr
RBF	LOGOUT or LOGOFF cr
IAF	BYE or LOGOUT or GOODBYE cr
MCS	BYE or LOGOUT cr
TAF	none
TVF	none
PNI	none
VEIAF	LOGOUT cr

For example, if your terminal is connected to RBF, you can disconnect from the host by entering the LOGOUT command.

DISCONNECTING WITH HOST SYSTEM COMMANDS

If your current application program does not have a command for direct disconnection, you must first disconnect your terminal from the current program with a command recognized by that application. Then you must request host disconnection from the host. Program disconnection commands are described earlier in this section under Switching to a Different Application.

After disconnecting the application program, the system indicates that a connection switch is in progress by issuing this message:

```
application - CONNECT TIME hh.mm.ss.
```

The application program that was just disconnected is indicated by application and the time that passed while the terminal was connected to the program is indicated by hh.mm.ss.

You then receive another prompt for an application program name:

```
terminalname - APPLICATION:
```

unless the terminal is permitted access to only one application program. If so, you are disconnected from the host anyway.

The following host disconnection commands are valid responses to the APPLICATION prompt on CDC host systems:

BYE cr

LOGOUT cr

For example, you can disconnect your terminal from RBF by entering the END command. You then receive a prompt for a new application name and enter BYE.

DISCONNECTING WITH A NETWORK SOFTWARE COMMAND

You normally end a connection with a host by using a command to disconnect you from an application and/or a command to log out of the host. Events might make it impossible for you to do either. When this happens, you can end your terminal's connection to the host by entering this terminal definition command:

ct TM cr

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

You can use the TM command if an application program does not respond to your entries or to initiate a connection to a different host.

Once the terminal has been disconnected from the host, you might want to reset all of the terminal characteristics to their original NDL settings. You use the RC command to do this. Its format is:

ct RC cr

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

This command affects only the console from which you enter it.

A terminal device can also be automatically reconfigured each time the terminal is disconnected from a host if it is on a line from which automatic reconfiguration has been specified by the network administrator.

RECONNECTING TO A HOST OR CONNECTING TO A DIFFERENT HOST

After your terminal is disconnected from the host, this message is printed:

```
HOST DISCONNECTED.  CONTROL CHARACTER = ct
```

followed by the prompting message:

```
ENTER INPUT TO CONNECT TO HOST
```

The ct parameter in this message is the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

You can then attempt another connection to the same host by sending any input you want. To reach a different host, you must complete the procedures for Selecting a Host System and Connecting to a Selected Host described earlier in this section.

INTERRUPTIONS

Events can sometimes interrupt your input or output. These interruptions fall into two categories:

- Interruptions for communication with the network operator
- Temporary suspensions in communication

COMMUNICATING WITH THE NETWORK OPERATOR

The term network operator includes both a host system operator and an NPU operator. Either operator can be a network operator and can send and receive messages that involve you.

Receiving Messages

The network operator can send short messages to a terminal. The message appears at your terminal in this format:

```
FROM NOP...message text
```

These messages can appear at any time a new line of output is possible at your terminal. For example, the network operator might want to inform all connected terminals that communication is going to be suspended intentionally with this message:

```
FROM NOP...SHUTDOWN IN 5 MIN
```

Sending Messages

You can respond to such messages by entering the following terminal definition command:

```
ct MS=message text  cr
```

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

The message text must not exceed 50 graphic characters. Because the network operator can be located at a terminal that does not support many special characters, you should avoid characters not in the ASCII 63-character subset described in appendix A.

You can also begin dialog with the network operator. For example, if you need to use a remote batch printer that has been logically disabled, the entry:

```
%MS=PLEASE ENABLE LP=M4C555  cr
```

causes the message PLEASE ENABLE LP=M4C555 to appear at the network operator's terminal, prefixed by the terminal name of your terminal. The % character represents the network control character for your terminal.

Preventing Messages

Sometimes it is important that output to your terminal does not include unplanned messages from the network operator. You can prevent delivery of messages from the network operator by locking them out with the following terminal definition command:

```
ct LK [ = { Y } ]  cr
          { N }
```

The ct parameter in this command must be the network control character defined for your terminal. The network control character is explained in section 5; refer to the CT command description.

Omitting the equals sign and a value has the same effect as entering a Y value.

After you enter this command with the value Y, any message to your terminal from the network operator is discarded by the network software. Messages are discarded until you again allow delivery by using the command with the value N.

Site administrators can configure your terminal so that messages are discarded as soon as the terminal is connected to the network. A network application program can also change the LK value in use.

SUSPENSIONS OF COMMUNICATION

Sometimes, message traffic in the network is so heavy that all storage is temporarily used up. Also, a host may be too busy to accept data. During these times, the network software takes steps that temporarily stop input until enough output has occurred to free additional storage.

Although storage may be low, sometimes the network cannot prevent you from entering data. Therefore, the network discards each message from your terminal and sends this message to your terminal:

```
WAIT..
```

When the following message appears, you can reenter any information that was previously discarded:

```
REPEAT..
```

If your terminal transmits more than one message at a time, you must determine which information you need to reenter. The application program might have commands that can help you determine which messages it received.

When only one block from a multiple-block message is discarded, the network sends this message to your terminal:

INPUT DISCARDED..

You should discard the remainder of the input affected and resend all of it.

This message can also appear when:

The network must discard batch data

When you enter a second priority data command, user-break-1 character, or user-break-2 character before the host processes the first one (see section 5)

When you enter input other than a terminal definition command before a connection to a host exists

CONNECTION FAILURES

Events can occur that:

Cause your application program to fail

Disconnect your terminal from an application program

Disconnect your terminal from the host

Disconnect your terminal from the network

APPLICATION PROGRAM FAILURES

If an application program fails after your terminal is connected, this message is displayed:

```
APPLICATION FAILED.
application CONNECT TIME hh.mm.ss.
terminalname - APPLICATION:
```

The length of time your terminal was connected to the application program is specified by hh.mm.ss; application is the name of the application program; and terminalname represents the terminal name the network uses to identify the terminal. You either can select another application or log out.

DISCONNECTION FROM AN APPLICATION AND/OR A HOST

An application program is informed of two events that you might want to take action on:

Long periods of device inactivity

Pending network shutdown

The network software informs an application if more than 10 minutes have elapsed without any communication between a device at your terminal and the application. An application program can:

Ignore this time interval

Ask you for continued dialog

Disconnect from your terminal or from the device with or without telling you

The effect of device inactivity depends on the application program with which the device is currently connected.

When shutdown of the network is pending, the network operator can enter a command that informs all application programs of the pending event. Your application program might take actions that result either in connection termination or application failure. It also might send you a message to warn you of the event.

If you are using a dial-up line, the network administrator might have selected the immediate disconnect option. As soon as the connection to the host is terminated, the line to the NPU is disconnected: you must dial back in.

If you receive a message from an application program indicating that shutdown is pending, you should take steps immediately to save all data or files you are currently using. Then end connection with both the application program and the host as soon as possible.

COMMUNICATION FAILURES

The following messages inform you that the network has lost communication with the host software; the host status message:

HOST DISCONNECTED. CONTROL CHARACTER = ct

followed by the prompting message:

ENTER INPUT TO CONNECT TO HOST

or the complete host availability display.

If you are at a dial-up terminal and you do not attempt to connect to the same or a different host within two minutes, your terminal is disconnected from the network.

If communication between the NPU and the host software resumes, you must connect to the host and log in again, but you do not need to access the network again (perform dial-up and auto-recognition procedures). These host communication failures occur either when the host has been shut down or when hardware fails in the communication path between the NPU and the host.

All network software conditions (that is, parameters that may be set with terminal definition commands) are retained across host disconnection except for transparent, full ASCII, and special edit input modes. See section 5 for a discussion of these commands.

Failures in the communication path between the network hardware and your terminal are treated by the network software as if your terminal had failed. These failures can occur because of hardware problems, or maybe because you entered an input character during output. If you hold the interactive output interrupt key down for too long, a communication failure can occur. This key is identified in section 2 for terminals of all classes.

There are several tests for communication failures. If your terminal has a CTS, ON-LINE, or CARRIER indicator, the light goes out when a failure occurs. At other terminals, if a failure has occurred, requesting the host availability display produces no response from the network.

Batch input and ouput devices associated with the failed console stop running, and batch device connections are ended. The network software does not distinguish between dial-up and hardwired terminals when processing terminal failures.

Terminal failures require you to complete new access and login procedures to resume communication with the network. Terminal failures always cause disconnection when the terminal is on a dial-up line.

The network software can support many kinds of terminals. This ability is provided by a portion of the network software called a Terminal Interface Program (TIP).

You are normally unaware that the Terminal Interface Program exists. There is a TIP for each major grouping of similar terminals (for each protocol, as listed in section 4). Each TIP performs similar functions, translating character codes used by the terminal into codes used by the host software with which you are communicating, collecting character-by-character input into lines of input to send to the host, and so forth.

The TIPs can only communicate with the devices of a specific terminal when all of that terminal's pertinent operating characteristics are known. The terminal's operating characteristics are made known to the TIP through a concept called a logical terminal. Each logical terminal comprises one or more logical devices. (Some documents call a logical device an interactive virtual terminal when the device is a console.)

When the site administrators configure the network, they define a logical terminal and a logical device to correspond to each real terminal and real device that can be active when the network is operating. Logical terminals are assigned to physical communication lines. Logical devices are assigned to logical terminals. Each logical device is assigned a terminal name, as described in section 4, and associated with a set of parameters that define its operating characteristics. These parameters are called terminal definition parameters.

Because many terminals have essentially the same operating characteristics, we can group a set of terminal definition parameters with specific default values and call the grouping a terminal class. The administrators therefore assign each logical terminal to an appropriate terminal class to establish all of the default values for its devices' operating characteristics. The administrators can change the characteristics for each device of the logical terminal from the defaults for the terminal class, but only within ranges and restrictions appropriate for the terminal class.

When a terminal physically connects to the network, it is associated with the logical terminal and the logical devices the TIP identifies for the communication line used. As far as the TIP is aware, the physical terminal and devices possess the operating characteristics of the logical terminal and devices.

USING TERMINAL INTERFACE PROGRAM TERMINAL DEFINITION COMMANDS

If the operating characteristics of the logical terminal's console are not completely appropriate for the physical terminal's console, you might select another communication line and associate the physical terminal with a different logical terminal. This is not always a practical or possible procedure.

Instead, you can enter terminal definition commands to change any of the terminal definition parameters associated with the logical terminal's console. You can also use a terminal definition command to change the terminal class of the logical terminal.

CDC defines eighteen terminal classes as having default values for operating characteristics. These predefined classes are numbered 1 through 18 and have the default values shown in tables 5-1 and 5-2. Terminal classes 19 through 27 are reserved by CDC for future definition. Classes 28 through 31 can be used by sites that create new terminal interface programs.

Each terminal class in tables 5-1 and 5-2 is associated with the type of terminal for which the default values were chosen as most appropriate. The following abbreviations are used in tables 5-1 through 5-4:

TTY	Teletype Corporation
Haz	Hazeltine Corporation
Tek	Tektronix Corporation teletypewriter emulator
X3.64	Terminals using the ANSI X3.64 standard
X.25	Terminals using the X.25 communication protocol through a packet assembly/disassembly (PAD) service
HASP	Terminals using the Houston Automatic Spooling Program (HASP) protocol
POST	Terminals using the HASP protocol, but supporting only postprint format control
PRE	Terminals using the HASP protocol, supporting preprint format control
UT	User Terminal

TABLE 5-1. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 1 THROUGH 8

Mnemonic	Description	Terminal Types								
		TTY M33	CDC 713-10, 722-10, 751-1, 752, 756	CDC 721	IBM 2741	TTY M40-2	Haz 2000	ANSI X3.64	Tek 4000	X.25 any
TC	Terminal Class	1	2	3	4	5	6	7	8	(0)
AB	Abort output block	CAN (X)	CAN (X)	CAN (X)	(ATTN (	CAN (X)	CAN (X)	CAN (X)	CAN (X)	na
AR	Automatic recognition code and character set	2	2	2	none	2	2	2	2	2
BF	Blocking factor	1	1	1	1	1	1	1	1	(1)
BS	Backspace character	BS (H)	BS <--	BS <--	BS BACK SPACE	none	BS (H)	BS BACK SPACE	BS (H)	(1)
B1	User break 1 character	DLE (P)	DLE (P)	DLE (P)	: ATTN :	DLE (F)	DLE (P)	DLE (P)	DLE (P)	(1)
B2	User break 2 character	DC4 (T)	DC4 (T)	DC4 (T)	) ATTN)	DC4 (T)	DC4 (T)	DC4 (T)	DC4 (T)	(1)
BR	Break as user break 1 (and as cancel input line)	N	N	N	na	N	N	N	N	(1) (na)
CI	Carriage return idle characters	2	0	0	8	1	0	0	0	(1)
CN	Cancel character	CAN (X)	CAN (X)	CAN (X)	(ATTN (	CAN (X)	CAN (X)	CAN (X)	CAN (X)	(1)
CP	Cursor positioning	Y	Y	Y	na	Y	Y	Y	Y	N
CT	Network control character	ESC ESC	ESC ESCAPE	ESC ESC	% ATTN %	ESC (P)	ESC ESC	% %	ESC ESC	(1)
DL	Single-message transparent input mode									
	Character delimiter	CR RETURN	CR RETURN	CR CR	NL RETURN	CR RETURN	CR CR	CR RETURN	CR CR	(1)
	Character count delimiter	2043	2043	2043	na	2043	2043	2043	2043	(1)
	Timeout delimiter	no	no	no	na	no	no	no	no	na

TABLE 5-1. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 1 THROUGH 8 (Contd)

Mnemonic	Description	Terminal Types								
		TTY M33	CDC 713-10, 722-10, 751-1, 752, 756	CDC 721	IBM 2741	TTY M40-2	Haz 2000	ANSI X3.64	Tek 4000	X.25 any
EB	End-of-block									
	Character	EOT (D)	EOT (D)	EOT (D)	na	EOT (D)	EOT (D)	EOT (D)	EOT (D)	(1)
	Cursor positioning	CL	CL	CL	na	CL	CL	CL	CL	NO
EL	End-of-logical line									
	Character	CR RETURN	CR RETURN	CR CR	NL RETURN	CR RETURN	CR CR	CR RETURN	CR CR	(1)
	Cursor positioning	LF	LF	LF	na	LF	LF	LF	LF	NO
EP	Echoplex mode	N	N	N	na	N	N	Y	N	(1)
FA	Full-ASCII input mode	N	N	N	N	N	N	N	N	(1)
IC	Input flow control	N	N	N	na	N	N	Y	N	(1)
IN	Input mechanism type	KB	KB	KB	KB	KB	KB	KB	KB	KB
LI	Line feed idle characters	1	0	0	1	3	3	0	0	(1)
LK	Lockout network operator messages	N	N	N	N	N	N	N	N	(1)
OC	Output flow control	N	N	N	na	N	N	Y	N	(1)
OP	Output mechanism type	PR	DI	DI	PR	DI	DI	DI	DI	(1)
PA	Parity processing	E	E	E	na	E	E	E	E	(2)
PG	Page waiting	N	N	N	N	N	N	N	N	(1)
PL	Page length	0	24	30	0	24	27	24	35	(1)
PW	Page width	72	80	80	132	80	74	80	74	(1)
SE	Special editing mode	N	N	N	N	N	N	N	N	(1)
XL	Multiple-message transparent input mode	none	none	none	none	none	none	none	none	none

TABLE 5-1. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 1 THROUGH 8 (Contd)

Notes:

Keys that generate network function codes appear beneath the ASCII character mnemonic for the code used.

A circled value means press the CTL, CTRL, CNTRL, or CONTROL key and the indicated key simultaneously. The ASCII codes generated by the keys should be the codes associated with the keys in appendix A.

CR or RETURN indicates the terminal key that causes a carriage return operation.

A zero page width or page length value is effectively an infinite one.

na Indicates the parameter does not apply to that terminal class.

CA indicates that a value calculated by the TIP is used.

(0) Devices belong to terminal classes 1 thru 3 or 5 thru 8.

(1) Same as for the device if it were not connected via a packet assembly/disassembly (PAD) service.

(2) The parity bit is ignored on input and is generated for output as if the device were not connected via a packet assembly/disassembly (PAD) service.

TABLE 5-2. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 9 THROUGH 18

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal Class	9	10	11	12	13	14	15	16	17	18
AB	Abort output block character	na	na	na	na	na	na	na	na	na	na
AR	Automatic recognition code and character set	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	na	1	1	1	1	na	1	na	na	1
BS	Backspace character	na	na	na	na	na	na	na	na	na	na
B1	User break 1 character	:	:	:	:	:	:	:	na	na	: (3)
B2	User break 2 character	)	)	)	)	)	)	)	na	na	) (4)
BR	Break as user break 1 (and as cancel input line)	na	na	na	na	na	na	na	na	na	na
CI	Carriage return idle characters	na	na	na	na	na	na	na	na	na	na
CN	Cancel character	(	(	(	(	(	(	(	na	na	(

TABLE 5-2. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 9 THROUGH 18 (Contd)

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
CP	Cursor positioning	na	na	na	na	na	na	na	na	na	na
CT	Network control character	%	%	%	%	%	%	%	%	%	%
DL	Single-message transparent input mode										
	Character delimiter	na	(1) SEND	(1) ETX	(1) ETX	(1) ETX	na	(1) ETX	na	na	ENTER
	Character count delimiter	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	na	na	na	na	na	na	na	na	na	na
EB	End-of-block										
	Character	na	(1) SEND	(1) ETX	(1) ETX	(1) ETX	na	(1) SEND	na	na	ENTER
	Cursor positioning	na	na	na	na	na	na	na	na	na	na
EL	End-of-logical line										
	Character delimiter	na	(2) RETURN	(2) NEW LINE	(2) NEW LINE	(2) NEW LINE	na	(2) NEW LINE	na	na	na
	Cursor positioning	na	NO	NO	NO	NO	na	NO	na	na	na
EP	Echoplex mode	na	na	na	na	na	na	na	na	na	na
FA	Full-ASCII input mode	na	N	N	N	N	na	N	na	na	N
IC	Input flow control	na	na	na	na	na	na	na	na	na	na
IN	Input mechanism type	KB	BK	BK	BK	BK	KB	BK	na	na	KB
LI	Line feed idle characters	na	na	na	na	na	na	na	na	na	na
LK	Lockout network operator messages	N	N	N	N	N	N	N	na	na	N
OC	Output flow control	na	na	na	na	na	na	na	na	na	na
OP	Output mechanism	na	na	na	na	na	na	na	na	na	na
PA	Parity processing	na	Odd	Odd	Odd	Odd	na	Odd	na	na	na
PG	Page waiting	na	Y	Y	Y	Y	na	Y	na	na	Y

TABLE 5-2. DEFAULT TERMINAL DEFINITIONS FOR TERMINAL CLASSES 9 THROUGH 18 (Contd)

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
PL	Page length	0	13	16	16	16	0	13	0	0	24
PW	Page width	80	80	80	80	80	80	80	80	120	80
SE	Special editing mode	na	na	na	na	na	na	na	na	na	na
XL	Multiple-message transparent input mode	na	none	none	none	none	na	none	na	na	na

Notes:

Keys that generate network function codes appear beneath the ASCII character mnemonic for the code used.

CR or RETURN indicates the terminal key that causes a carriage return operation.

A zero page width or page length value is effectively an infinite one.

na Indicates the parameter does not apply to that terminal class.

(1) ETX (resulting from the SEND key).

(2) Escape sequence (ESC and A, resulting from the carriage return key).

(3) The program attention 1 key is a valid user break 1.

(4) The program attention 2 key is a valid user break 2.

TABLE 5-3. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 1 THROUGH 8

Mnemonic	Description	Terminal Types								
		TTY M33	CDC 713-10, 722-10, 751-1, 752, 756	CDC 721	IBM 2741	TTY M40-2	Haz 2000	ANSI X3.64	Tek 4000	X.25 any
TC	Terminal class	1	2	3	4	5	6	7	8	(0)
AB	Abort output block character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na
AR	Automatic recognition code and character set	2 thru 4	2 thru 4	2 thru 4	5 thru 8	2 thru 4	2 thru 4	2 thru 4	2 thru 4	(2)
BF	Blocking factor	0 thru 2	0 thru 2	0 thru 2	0 thru 2	0 thru 2	0 thru 2	0 thru 2	0 thru 2	(2)
BS	Backspace character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)
B1	User break 1 character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)

TABLE 5-3. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 1 THROUGH 8 (Contd)

Mnemonic	Description	Terminal Types								
		TTY M33	CDC 713-10, 722-10, 751-1, 752, 756	CDC 721	IBM 2741	TTY M40-2	Haz 2000	ANSI X3.64	Tek 4000	X.25 any
B2	User break 2 character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)
BR	Break as user break 1 (and as cancel input line)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	na na	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	Y,N (Y,N)	(2) (na)
CI	Carriage return idle characters	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	(2)
CN	Cancel character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)
CP	Cursor positioning	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)
CT	Network control character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(2)
DL	Single-message transparent input mode									
	Character delimiter	0 thru FF	0 thru FF	0 thru FF	na	0 thru FF	0 thru FF	0 thru FF	0 thru FF	(2)
	Character count delimiter	1 thru 4095	1 thru 4095	na	na	1 thru 4095	1 thru 4095	1 thru 4095	1 thru 4095	(2)
	Timeout delimiter	TO	TO	TO	na	TO	TO	TO	TO	(2)
EB	End-of-block									
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	NL	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)
	Cursor positioning	(3)	CR	CR	na	(3)	(3)	(3)	CR	(2)
EL	End-of-logical line									
	Character	(1), EL, EB	(1), EL, EB	(1), EL, EB	na	(1), EL, EB	(1), EL, EB	(1), EL, EB	(1), EL, EB	(2)
	Cursor positioning	(3)	(3)	(3)	na	(3)	(3)	(3)	(3)	NO
EP	Echoplex mode	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)
FA	Full-ASCII input mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)

TABLE 5-3. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 1 THROUGH 8 (Contd)

Mnemonic	Description	Terminal Types								
		TTY M33	CDC 713-10, 722-10, 751-1, 752, 756	CDC 721	IBM 2741	TTY M40-2	Haz 2000	ANSI X3.64	Tek 4000	X.25 any
IC	Input flow flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)
IN	Input mechanism type	(4)	(4)	(4)	KB,XK, X	(4)	(4)	(4)	(4)	BK,KB X,XK
LI	Line feed idle characters	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	0 thru 127, CA	(2)
LK	Lockout network operator messages	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)
OC	Output flow control	Y,N	Y,N	Y,N	na	Y,N	Y,N	Y,N	Y,N	(2)
OP	Output mechanism type	(5)	(5)	(5)	PR,DI	(5)	(5)	(5)	(5)	PR,DI
PA	Parity processing	Z,O,I, E,N	Z,O,I, E,N	Z,O,I, E,N	na	Z,O,I, E,N	Z,O,I, E,N	Z,O,I, E,N	Z,O,I, E,N	(6)
PG	Page waiting	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	(2)
PL	Page length	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	(2)
PW	Page width	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	(2)
SE	Special editing mode	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na
TC	Terminal class, input	1, 2, 3, 5 thru 8	1, 2, 3, 5 thru 8	1, 2, 3, 5 thru 8	4	1, 2, 3, 5 thru 8	1, 2, 3, 5 thru 8	1, 2, 3, 5 thru 8	1, 2, 3, 5 thru 8	(1)
XL	Multiple-message transparent input mode									
	Character delimiter	0 thru FF	0 thru FF	0 thru FF	na	0 thru FF	0 thru FF	0 thru FF	0 thru FF	(2)
	Character end-of-message indicator	0 thru FF	0 thru FF	0 thru FF	na	0 thru FF	0 thru FF	0 thru FF	0 thru FF	(7)
	Timeout delimiter	TO	TO	TO	na	TO	TO	TO	TO	na
	Character-count delimiter	1 thru 4095	1 thru 4095	1 thru 4095	na	1 thru 4095	1 thru 4095	1 thru 4095	1 thru 4095	(2)

TABLE 5-3. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 1 THROUGH 8 (Contd)

Notes:

CR or RETURN indicates the terminal key that causes a carriage return operation.

A zero page width or page length value is effectively an infinite one.

na Indicates the parameter does not apply to that terminal class.

CA Indicates that a value calculated by the TIP is used.

(0) Devices belong to terminal classes 1 thru 3, or 5 thru 8.

(1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.

(2) Same as for the device if it were not connected via a packet assembly/disassembly (PAD) service.

(3) CR, LF.

(4) KB, PT, BK, X, XK, XP.

(5) PR, DI, PT.

(6) Same as if the device were not connected via a packet assembly/disassembly (PAD) service. Parity is ignored on input and is generated for output.

(7) PAD subTIP devices restricted to characters in columns 0 and 1 of the ASCII character set (0-1F); See appendix A.

TABLE 5-4. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 9 THROUGH 18

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
TC	Terminal class	9	10	11	12	13	14	15	16	17	18
AB	Abort output block character	na	na	na	na	na	na	na	na	na	na
AR	Automatic recognition code and character set	na	na	na	na	na	na	na	na	na	na
BF	Blocking factor	na	0-2	0-2	0-2	0-2	na	0-2	na	na	na
BR	Break as user break 1 (and as cancel input line)	na	na	na	na	na	na	na	na	na	na
BS	Backspace character	na	na	na	na	na	na	na	na	na	na
B1	User break 1 character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
B2	User break 2 character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
CI	Carriage return idle characters	na	na	na	na	na	na	na	na	na	na
CN	Cancel character	(1)	(1)	(1)	(1)	(1)	(1)	(1)	na	na	(1)
CP	Cursor positioning	na	na	na	na	na	na	na	na	na	na

TABLE 5-4. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 9 THROUGH 18 (Contd)

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
DL	Single-message transparent input mode										
	Character delimiter	na	03	03	03	03	na	03	na	na	na
	Character count delimiter	na	na	na	na	na	na	na	na	na	na
	Timeout delimiter	na	na	na	na	na	na	na	na	na	na
EB	End-of-block										
	Character	na	(2)	(2)	(2)	(2)	na	(2)	na	na	na
	Cursor positioning	na	na	na	na	na	na	NO	na	na	na
EL	End-of-logical line										
	Character	na	(3)	(3)	(3)	(3)	na	(3)	na	na	na
	Cursor positioning	na	NO	NO	NO	NO	na	na	na	na	na
EP	Echoplex mode	na	na	na	na	na	na	na	na	na	na
FA	Full-ASCII input mode	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na	na
IC	Input mechanism flow control	na	na	na	na	na	na	na	na	na	na
IN	Input mechanism type	KB	X,BK	X,BK	X,BK	X,BK	KB	X,BK	na	na	KB
LI	Line feed idle characters	na	na	na	na	na	na	na	na	na	na
LK	Lockout network operator messages	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	Y,N	na	na	Y,N
OC	Output flow control	na	na	na	na	na	na	na	na	na	na
OP	Output mechanism type	DI	DI	DI	DI	DI	DI	DI	PR	PR	DI
PA	Parity processing	na	na	na	na	na	na	na	na	na	na
PG	Page waiting	na	Y,N	Y,N	Y,N	Y,N	na	Y,N	na	na	Y,N
PL	Page length	na	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	0, 8 thru 255	na	0, 8 thru 255	na	na	8 thru 255
PW	Page width	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	0, 20 thru 255	20 thru 255
SE	Special editing mode	na	na	na	na	na	na	na	na	na	na
TC	Terminal class, input	9	10, 15	11 thru 13	11 thru 13	11 thru 13	14	10, 15	16	17	18

TABLE 5-4. PARAMETER RANGES FOR TERMINAL DEFINITION COMMANDS, TERMINAL CLASSES 9 THROUGH 18 (Contd)

Mnemonic	Description	Terminal Types									
		HASP POST	CDC 200 UT	CDC 714-30	CDC 711-10	CDC 714-10, 714-20	HASP PRE	CDC 734	IBM 2780	IBM 3780	IBM 3270
XL	Multiple-message transparent input mode										
	Character delimiter	na	03	03	03	03	na	03	na	na	na
	Character end-of-message indicator	na	(3)	(3)	(3)	(3)	na	(3)	na	na	na
	Timeout delimiter	na	na	na	na	na	na	na	na	na	na
	Character-count delimiter	na	na	na	na	na	na	na	na	na	na

Notes:

CR or RETURN indicates the terminal key that causes a carriage return operation.

A zero page width or page length value is effectively an infinite one.

na Indicates the parameter does not apply to that terminal class.

(1) Any character from the ASCII 128-character set except any lowercase or uppercase alphabetic character, any digit 0 through 9, NULL, SOH, STX, =, DEL, or space.

(2) A sequence resulting from pressing the SEND key.

(3) A sequence resulting from pressing the CARRIAGE RETURN key or SEND key.

Not all terminal definition commands involve the physical operation of the terminal. Some commands alter host software operations; other commands (described in section 4) control network functions.

Some commands might not have any detectable effect, or have effects that depend on the network application program currently being used. Such operations as reading or punching paper tape, paging displays, suppressing TIP editing of input, and formatting display or printer output might require the use of terminal definition commands.

Most of the terminal definition commands available to you can also be sent downline by an application program. Commands you enter are removed from the input data by the TIP and acted on directly. A network application program is advised of a terminal definition command entry only when the command changes one of three terminal characteristics:

Terminal class

Page width (number of characters potentially output per physical line)

Page length (number of lines of the specified page width treated as a unit when dividing output into pages)

All other terminal definition commands either are of no consequence to an application program or have implicit effects on the input data.

Terminal definition commands sent downline by a network application program are removed from the output stream by the TIP and acted on directly. You are not advised of their occurrence in the output stream.

The legal range of values within each terminal class that you can assign to command keywords is shown in tables 5-3 and 5-4. If your network uses terminal interface programs other than those supporting terminal classes 1 through 18, an administrator must provide you with corresponding range information.

COMMAND FORMAT, SYNTAX, AND CONSTRAINTS

This subsection describes the general format and syntax of all terminal definition commands. The specific format for each command concerned with changing the terminal interface is described separately in this section. Other terminal definition commands (HD, HN, HS, LK, MS, and TM) are described in section 4.

Constraints that apply to all terminal definition command uses are described below. Constraints, ranges, and dependencies unique to a specific command, keyword, or value are described in the subsection describing that command or keyword.

GENERAL FORMAT AND SYNTAX

You can change the terminal class or operating characteristics assigned to a logical terminal by entering a terminal definition command with the general format and syntax shown in figure 5-1. You enter the command in one of three formats, depending on the specific command:

keyword

keyword=value

value

A keyword is always a two-character mnemonic for the command's function. A value can be one or more ASCII characters or a numeric value.

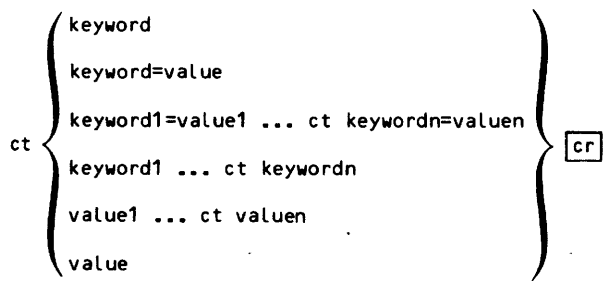
Most of the commands use the keyword=value form. For some commands, the value portion is optional and has a default described in the text. For other commands, no default exists and you must supply a value within the range indicated for the command or keyword.

For example, to change the page width of a 200 User Terminal, type the command:

%PW=50

and press the SEND key. In this example, the % character is the network control character defined for the terminal. The characters PW=50 are the command itself, where PW is the keyword mnemonic for the page width, and 50 is the new page width value (a decimal number of characters) for the terminal console. The PW keyword has no default value, so you must enter a number in the legal range for the PW keyword. The SEND key causes logical line and block transmission from a 200 User Terminal.

The % character is the default network control character for many terminal classes. This character is used in examples throughout this guide.



ct	Represents the network control character for your device as defined by the CT terminal definition command. Tables 5-1 and 5-2 give the default network control character for your device. You must press the ATTN key on a 2741-compatible device before entering this character.
keyword	Represents a two-character mnemonic associated with a device characteristic or a network software function.
value	Specifies a particular value associated with the terminal definition command mnemonic. Tables 5-3 and 5-4 give the range of values for each command keyword. In some commands, the =value portion of the command is optional and defaults to =Y , representing a selection of yes.
keywordi	Same as keyword.
valuei	Same as value.
cr	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

NOTE: The spaces shown are for clarity only. Do not include spaces when you enter a terminal definition command.

Figure 5-1. Functional Formats of Terminal Definition Commands

CONSTRAINTS

At block mode terminals (see section 2), the terminal definition command or commands should be the last input line in the transmission block. If the input is from paper tape, the tape reader should stop after the line is entered. If more input follows a terminal definition command in the transmission block, the additional input is discarded and a

```
REPEAT..
```

message is sent to the console.

The network responds to a valid terminal definition command with:

```
xx ACCEPTED..
```

The command keyword is represented by xx.

The network responds to an invalid command with one of several diagnostic messages described in appendix B.

You should not enter terminal definition commands or characters that flag TIP functions while output is in progress. Such entries are often not correctly recognized by the TIP and are therefore transmitted as input data to the application program. You can safely make such entries when the terminal provides a hardware mechanism that interrupts output without generating a character of data in the process. You can also safely make such entries when output pauses because page waiting is in use.

You can enter many of the commands independently; entry of one command usually does not change more than one operating characteristic or affect the use of other characteristics. There are exceptions to this, however.

Changing the terminal class changes all of the characteristics associated with the terminal from their previous values to the default values for the new terminal class. The original, preset configuration of the terminal cannot be recovered unless you reconstruct it by subsequent terminal definition commands, or the network is reinitialized, or (at a dialup terminal) the terminal is reconnected to the network.

Neither logout nor switching application programs changes or restores terminal definition parameter values. (Disconecting your terminal from a host does end transparent, full-ASCII, or special editing mode operation, however.)

If you want to reset the terminal characteristics to the values defined in the network configuration file, use the RC command described in section 4.

You should use terminal definition commands carefully. You should not change terminal class without strong justification.

You should also be careful when using commands that associate terminal functions with input of a specific character. A terminal should never have two functions associated with a single character, either explicitly through terminal definition commands, or implicitly; that is, you should avoid changes that would render a terminal functionally inoperable.

Any character or character code you use as a value in a terminal definition command is subject to the following restrictions:

- You cannot use any of the following characters as an indicator for a TIP function:

ASCII Character	Hexadecimal ASCII Code
NUL	00
SOH	01
STX	02
space	20
=	3D
DEL	7F

  Neither these characters, nor their corresponding values, nor any terminal characters or codes translated to these characters, can follow the required equal sign in the command format.

- Each character flagging a TIP function must be unique for a given terminal. You cannot use the same character or character code as a value in two or more of the following commands:

  AB or CN, BS, B1, B2, CT, EB or EL

- If the terminal does not use the ASCII character set, you should choose a character that has a unique translation to a character in the 7-bit ASCII code set.

- You cannot use a character or character code as a value in a terminal definition command if it already has been used explicitly (or as a default value) in another command for the same console. There are exceptions to this restriction: the abort-output-block character and cancel-input character commands (AB and CN) can use the same value, and the end-of-logical-line character and end-of-block character commands (EL and EB) can use the same value.

- You cannot use a character edited from input as an indicator character for a TIP function. Neither the following characters, nor the ASCII codes corresponding to these characters, nor terminal characters or character codes translated to these ASCII codes, should follow the required equal sign in the keyword=value format when editing would remove them from the input:

ASCII Character	Hexadecimal ASCII Code
EOT	04
ETX	03
CR	0D
LF	0A
DC1	11
DC3	13

  Specifically, you cannot use DC1 or DC3 as a value in the following commands when output control is in effect (OC command Y option): AB or CN, BS, B1, B2, CT, EB or EL.

- You cannot use an uppercase or lowercase letter or a numeral as an indicator character for a TIP function performed during input. Neither these characters, nor terminal characters or character codes translated to these ASCII codes, can occur in the commands AB, B1, B2, BS, CN, CT, EL, or EB:

ASCII Character	Hexadecimal ASCII Code
0 thru 9	30 thru 39
A thru Z	41 thru 5A
a thru z	51 thru 7A

  A letter can be used as a priority data character in the command that bypasses queued input. The priority data command is described later in this section.

Some terminal definition commands are allowed only from terminals that use certain communication protocols. Other commands are allowed only from terminals in certain terminal classes. Each of the CDC-supported protocols includes one or more CDC-defined terminal classes, as follows:

Asynchronous (terminal classes 1 through 8)

X.25 packet-switching network (PSN) packet assembly/disassemby (PAD) service terminals, which are a subset of asynchronous terminals (terminal classes 1 through 3 and 5 through 8)

Mode 4 (terminal classes 10 through 13 and 15)

HASP (terminal classes 9 and 14)

Bisynchronous (terminal classes 16 and 17)

3270 (terminal class 18)

TERMINAL DEFINITION COMMAND KEYWORDS AND VALUES

The terminal definition keyword and value descriptions in the rest of this section explain the substitutions allowed for the value parameter in the command format of figure 5-1. The keyword or keyword and value represented by termdef is used to change terminal class or assigned terminal definition values.

Default values for terminal characteristics are not shown in the keyword descriptions because they vary with the terminal class. See tables 5-1 and 5-2 for default values. If you cannot find the key on your keyboard that generates a particular character in these tables, refer to the operator's guide for your terminal. Some of the tables in appendix A of this book also show letter keys that can often be pressed with a CONTROL key to generate certain ASCII character codes.

DISPLAYING COMMAND VALUES IN USE

Before you change any of the terminal definition values in use, you might want to determine what those values are. Two commands give you this information:

The HD command, described in section 4, tells you what network control character is defined for your device; this information appears in the first line of the display the command produces. (The network control character is explained as part of the CT command description later in this section.)

The CH command, described in the following subsection, produces a display of the most frequently used TIP function values that you can change.

Displaying Device Characteristics (CH)

The characteristics (CH) command can be entered to display a single line of information about the device's currently defined TIP function values. The format of the CH terminal definition command is given in figure 5-2.

```
ct CH [cr]

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.
        You must press the ATTN key on a
        2741-compatible device before entering
        this character.

[cr]    Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-2. Display Terminal Characteristics (CH) Command

This command is valid from console devices in terminal classes 1 through 18. The CH command cannot be sent downline by an application program.

When you enter this command, the TIP sends a line in the following format to your console:

```
TC=tc,BS=bs,CN=cn,AB=ab,B1=b1,B2=b2,EL=el,EB=eb
```

This line indicates the values in effect for the terminal definition commands TC, BS, CN, AB, B1, B2, EB, and EL described later in this section. The variable portions of the line appear as graphic characters if printable, as ASCII code mnemonics if not printable, or as N/A if the characteristic is not applicable to your device.

For example, suppose you are sitting at a CDC 721 Viking Extended terminal and your device uses the network control character %. You enter the following command:

```
%CH [cr]
```

You might see the following line:

```
TC=3,BS=(BS),CN=(CAN),AB=(CAN),B1=#,B2=),EL=(CR),
EB=(CR)
```

This line tells you that your device belongs to terminal class 3, uses the ASCII CAN character for both the cancel-input and abort-output-block functions, uses a # for a user-break-1 character, uses a) for a user-break-2 character, and uses a carriage return code for both the end-of-line and end-of-block functions.

CHANGING COMMAND RECOGNITION

Two terminal definition commands can be used to alter the recognition of all other commands by the TIP. The terminal class (TC) command changes the defaults,ranges, and values in effect for all terminal definition commands and TIP functions your device can use. A command-to-TIP (CT) command changes the network control character for the device and determines which entries from the terminal are recognized as terminal definition commands.

Changing Terminal Class (TC)

The terminal class (TC) command defines a value from 1 to 31 that associates the terminal with a set of operating characteristics. This aspect of the terminal class was discussed at the beginning of this section.

The terminal class associated with the terminal determines the following:

- the terminal definition commands that can be legally entered from the terminal console
- the default values of unaltered terminal definition parameters for the terminal
- the ranges permitted for all legal terminal definition commands from the terminal

The eighteen terminal classes defined by CDC have default values for terminal definition parameters. These predefined classes, numbered 1 through 18, have the default values shown in tables 5-1 and 5-2. Terminal classes 19 through 27 are reserved by CDC for future definition; classes 28 through 31 can be used by sites that create new terminal interface programs. The terminal class and command descriptions in this manual are limited to those defined by CDC.

The format of the terminal definition command to change the terminal class is given in figure 5-3. Tables 5-3 and 5-4 list the ranges for the terminal definition parameters of the new terminal class specified in the command.

This command is valid from console devices in terminal classes 1 through 18. The TC command can also be sent downline by an application program.

When you change the terminal class, the TIP changes all of the characteristics associated with the terminal from their previous values to the default values for the new terminal class. Previous values cannot be restored without entering a command for each corresponding terminal definition parameter.

Any previous value not restored by a subsequent terminal definition command will be restored when the terminal is disconnected from the network and then reconnected with it. The terminal configuration process causes all terminal definition parameters associated with the terminal to be preset when the communication path with the network software is established. The preset values are chosen by the site administrators.

The method used to establish a communication path can include automatic recognition of some terminal characteristics by the TIP. For example, a communication line can be configured so that the TIP automatically assigns the terminal to a terminal class. The TIP chooses the most appropriate value from the set of six possibilities 1, 4, 9, 10, 13, or 16. This set includes one value from each of the communication protocol variants supported.

If the assigned terminal class is not the most appropriate class for the actual terminal, you can use a terminal definition command to change it. You should choose a new terminal class whose default terminal definition corresponds most closely to the characteristics of your device. You cannot, however, choose a terminal class that requires a subprotocol different from that you are currently using.

Because the effect of the TC command is broad, this command should be the first one you enter if more than one needs to be entered. For example, if you want to be recognized as using a 734 terminal with a page width of 72 characters, but you are automatically recognized as using a CDC 200 User Terminal with a page width of 50 characters, the command sequence:

```
%TC=15  [cr]
%PW=72  [cr]
```

causes the desired redefinition. However, if you enter the commands in the opposite order, the terminal is left with a page width value of 80 characters, because entry of the TC parameter after the PW parameter changes the page width value in effect to the default for terminal class 15.

Changing Network Control Character (CT)

The command-to-TIP (CT) parameter defines a character used by the TIP to identify terminal definition commands at the beginning of input transmissions from your console. When this network control character is encountered at the beginning of a logical line, the TIP interprets the data as a command or as priority data for the connected host application program. When the TIP identifies a terminal definition command, it removes the data following the character from the line and acts upon it.

ct TC=tc2 [cr]

ct Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

tc2 Specifies the new terminal class, in the same range as your device's current terminal class (tc1), as follows:

If $1 \leq tc1 \leq 3$ or $5 \leq tc1 \leq 8$, then your device is an asynchronous terminal or connected through an X.25 packet-switching network, and tc2 can have the values:

$1 \leq tc2 \leq 3$ and $5 \leq tc2 \leq 8$

If tc1=4, then your device is a 2741-compatible asynchronous terminal, and tc2 can have the value 4.

If tc1=9 or tc1=14, then your device is part of a HASP terminal, and tc2 can only have the same value as tc1.

If tc1=10 or tc1=15, then your device is part of a mode 4A terminal, and tc2 can have the values 10 or 15.

If $11 \leq tc1 \leq 13$, then your device is part of a mode 4C terminal, and tc2 can have the values:

$11 \leq tc2 \leq 13$.

If $16 \leq tc1 \leq 18$, then your device is part of a bisynchronous terminal, and tc2 can only have the same value as tc1.

If $28 \leq tc1 \leq 31$, then your device is site-defined, and valid values for tc2 are unknown; possible values are:

$1 \leq tc2 \leq 18$ and $28 \leq tc2 \leq 31$.

The following list identifies the terminals that determine the default values for the characteristics of each terminal class:

tc2	Description
1	Teletype Model 33, 35, 37, or 38
2	CDC 713, 722-10, 751-1, 752, or 756 display terminal
3	CDC 721 Viking Extended display terminal or CDC CYBER 110 personal computer
4	IBM 2741 teletypewriter
5	Teletype Model 40-2 display terminal
6	Hazeltine 2000 series display terminal
7	Digital Equipment Corporation VT100 display terminal (ANSI X3.64)
8	Tektronix 40xx series teletypewriter-emulator display terminal
9	HASP terminal supporting only postprint format control
10	CDC 200 User Terminal
11	CDC 711-30/40 mode 4C terminal
12	CDC 711-10 mode 4C terminal
13	CDC 714-10/20 mode 4C terminal
14	HASP terminal supporting preprint format control
15	CDC 731-12 or 734 mode 4A terminal
16	IBM 2780 remote batch station
17	IBM 3780 remote batch station
18	IBM 3270 terminal
28 thru 31	Site-defined device

[cr] Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-3. Terminal Class (TC) Command

You must use a character in the 128-character ASCII character set as the network control character, with the restrictions described at the beginning of this section. The format of the terminal definition command to change the network control character is given in figure 5-4. If you enter the CT command in the same line as other terminal definition commands, the new value for the control character is not used until the entire line is processed.

```
ct CT=char [cr]

ct      Represents the current network control
        character for your console. You must
        press the ATTN key on a 2741-compatible
        device before entering this character.

char    Specifies the new network control
        character. This character must differ
        from the characters currently defined
        for the AB, BS, B1, B2, CN, EB, and EL
        command functions.

[cr]    Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-4. Network Control Character (CT) Command

This command is valid from console devices in terminal classes 1 through 18. The CT command can also be sent downline by an application program. As shown in tables 5-1 and 5-2, the default character varies according to terminal class.

You would normally change the network control character in situations where the character defined for the command flag function regularly occurs as the first line of a transmission. For example, if a text editor program requires you to use % as a tab character, you will often enter the % character from a 714 terminal as the first character of a line.

To avoid losing data and causing spurious error messages from the TIP, your site should either configure the terminal to use a flag character such as !, or require entry of the following from all 714 terminals before beginning text editor operations:

```
%CT=! [cr]
```

After this entry, lines beginning with % are passed by the TIP as normal data, and lines beginning with ! are scanned by the TIP as terminal definition commands. To change the network control character back to the default without leaving and rejoining the network, you would enter the following:

```
!CT=% [cr]
```

CONTROLLING INPUT OR OUTPUT CONTENT

The Terminal Interface Program allows you to edit input content before transmission to the host, and can be used in a limited way to perform input editing after transmission to the host. Similarly, you can use the TIP to perform limited editing of output in progress. These editing functions are performed as a response to recognition of selected character codes by the TIP, as described in section 2.

The TIP associates character codes with each of the following functions:

- Cancel the previous input character (a backspace function).
- Cancel the current or previous input line.
- Abort the current network output block transmission.

A terminal definition command containing the backspace (BS) parameter changes the character associated with the first operation. A command containing the cancel-input (CN) parameter changes the character associated with the second operation, and a command containing the abort-output-block (AB) parameter redefines the character that causes the third operation.

Changing Backspace Character (BS)

The TIP can edit input data to delete characters when the terminal is not operating in transparent mode, full-ASCII mode, or special editing mode (see section 2). Each entry of the character defined as the backspace character causes the TIP to remove the preceding character entry from its input buffer. The backspace character entered is not part of the input data.

This input editing operation permits you to replace an incorrect character in the input line with a correct character, as described in section 2. The backspace (BS) terminal definition command permits you to change the character used for this backspacing operation.

You must use a character in the 128-character ASCII character set as a backspace character, with the restrictions described at the beginning of this section. The format of the command to change the backspace character is given in figure 5-5.

The BS command is valid only for console devices in classes 1 through 8 (asynchronous and X.25 terminals). This command can also be sent downline for devices in those terminal classes by an application program.

As shown in tables 5-1 and 5-2, the default character for all terminal classes is generated by a key labeled with either a graphic character indicating the backspace function, or an alphabetic indicator for that function. Alternatively, at most terminals you can press a CONTROL key and a character key at the same time to perform the function.

ct BS=bs [cr]

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
bs	Specifies the new backspace character. This character must differ from the characters currently defined for the AB, B1, B2, CN, CT, EB, and EL command functions.
[cr]	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-5. Backspace Character (BS) Command

Regardless of the character used to identify the backspace character at the terminal, all of the default characters correspond to the ASCII code for the ASCII character BS (hexadecimal 08). The Model 40 Teletype has no default because no key generates this character.

You could use the BS command to create a backspace character for consoles in terminal class 5. You could also change the backspace character in situations where the ASCII backspace character code is part of the data required by the network application program communicating with a console in any terminal class between 1 and 8.

For example, some interpreters use the backspace character code to create a language input character by logically overstriking the character preceding the backspace character code with the character following the backspace character code. By changing the character associated with the backspace function from the character associated with the backspace code, you can functionally edit input lines by backspacing and still send a backspace code to an interpreter package.

The following sequence of entries illustrates the distinction between backspace code and backspace function. If the terminal uses a key marked <-- to generate a code which the network translates into the ASCII code for the backspace character, then the following entry causes an application program to receive the three characters ABC:

A Q <-- B) <-- C [cr]

However, the following entries cause the TIP to distinguish between a character for the backspace function and the backspace character code, causing the same entry line to be received by an application program as the five characters A Q <-- <-- C:

%BS=) [cr]

A Q <-- B) <-- C [cr]

In this second sequence of entries, the TIP has erased the B from its input buffer after receiving the new backspace character), and also discarded the) from the input line. The TIP has continued to edit the input line, but has changed the character it associates with the backspace function.

At line mode or block mode terminals (which support off-line input editing before transmission), the example given above does not work. Instead, changing the backspace character used by the TIP effectively produces two backspace characters and results in input editing at two points:

Editing in the terminal before transmission to the NPU, based on the backspace function associated with the original backspace character.

Editing in the network processing unit before transmission to the host, based on the backspace function associated with the redefined backspace character.

If two points of input editing exist, the entries just shown would cause the TIP to receive the three characters ABC and transmit them to the host without additional editing.

Unless you declare an SE value or FA value of Y before entering the backspace character from the terminal, each entry of the backspace character deletes another previously entered character from the terminal's input buffer. No data is sent upline to the host application program.

When the page width is greater than 56 characters, you can backspace the console to the beginning of the current physical line, but additional backspace character entries have no effect. The current physical line begins after the last line feed or carriage return, or at the character after you last reached the page width in the current logical line.

When the page width is greater than 20 but less than 56 characters, you can backspace into a previous physical line if that line has not already been sent to the host. You cannot backspace into a different network data block.

When you use a page width of 0 characters, input characters are sent to the host in multiples of 100 characters (see the PW command). You cannot backspace across these network data block boundaries.

Changing Cancel-Input Character (CN)

If you need to correct many characters of an input line, or if you decide a completely different entry should be transmitted as input to a network application program, an easier method than multiple back spaces can be used. If you enter a character defined as the cancel-input character and then press [cr], the TIP tells the host application program that it should discard the entire input message, including parts already received. The TIP then sends the message:

DEL

to your console and positions the cursor at the beginning of the next physical line.

Unless you declare an FA value of Y before entering the cancel-input character, each entry of this character at the end of an input line causes the character and the logical line preceding it to be discarded. Entry of the cancel-input character does not cause data transmission.

If part of the input line has been sent to the host, the TIP cannot perform a backspace operation to correct characters in it, or erase the entire line. However, you can enter a cancel-input character as the last character of the next transmission. This entry causes the TIP to request discarding of the entire message received by the host software. The action taken then depends on the network application program communicating with the terminal. Normally, the message is discarded unless processing of it cannot be reversed, and you can enter a corrected version of the message.

You cannot use the cancel-input character to satisfy a requirement for transmission to the host. You cannot enter the cancel-input character to satisfy the requirement to enter input after an automatic input prompt is output. Even though the automatic input prompt characters have been discarded, you must enter input before any more output can occur.

At an IBM-2741-compatible terminal, you must press the ATTN key after typing the cancel-input character.

The cancel-input (CN) terminal definition command permits you to change the character used for this cancel-input operation. This command is valid from console devices in terminal classes 1 through 15 and 18 (asynchronous, X.25, mode 4, HASP, and 3270 terminals). The CN command can also be sent downline for devices in those terminal classes by an application program.

You must use a character in the 128-character ASCII character set as a cancel-input-line character, with the restrictions described earlier in this section. The format of the terminal definition command to change the cancel-input character is given in figure 5-6. As shown in tables 5-1 and 5-2, the default character varies according to terminal class.

ct CN=cn [cr]

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
cn	Specifies the new cancel-input character. This character must differ from the characters currently defined for the BS, B1, B2, CT, EB, and EL command functions.
[cr]	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-6. Cancel-Input Character (CN) Command

You would normally change the cancel-input character in situations where the last character needed in many lines of data is the character currently defined for the function. For example, suppose the $ character is defined by your site as the cancel-input character used by all terminals. The $ is used in many programming functions as a separator or terminator character; if you are involved in such programming, you often end a line with this character. So that you need not remember to type a blank after the $ before pressing [cr], you can change the cancel character from $ to @ by entering the following command:

%CN=@ [cr]

This example assumes that the site defines the network control character as % for your terminal.

After making the entry, you no longer need to worry about inadvertently deleting a line of valid data by typing the $ before the transmission key is pressed. Only entries ending in the character @ are deleted.

Changing Abort-Output-Block Character (AB)

You sometimes encounter situations where unwanted or unanticipated output operations begin. You do not want to wait for all of the output to be completed before entering input, or do not want the extraneous output appearing on the hardcopy for the terminal session. The TIP provides a method of suppressing such output, called the abort-output-block character.

When you enter the abort-output-block character from the terminal as the only character in a line, the transmission block (one or more physical lines) currently being sent to the terminal is discarded.

The abort-output-block character is not sent to the application program as input data. If more than one block of output has been stored for transmission to the terminal, you must enter the character once for each of the blocks to be discarded.

The abort-block (AB) terminal definition command permits you to change the character used for this abort-output operation. This command is valid from console devices in terminal classes 1 through 8 (asynchronous terminals) when the terminals are not connected through an X.25 network. The AB command can also be sent downline to devices in those terminal classes by a network application program.

You must use a character in the 128-character ASCII character set as an abort-output-line character, with the restrictions described earlier in this section. The format of the terminal definition command to change the abort-output-block character is given in figure 5-7. As shown in tables 5-1 and 5-2, the default character varies according to terminal class.

You would normally change the abort-output-block character when the character used for that function is more appropriately associated with another function. For example, suppose you are using a CDC 713 terminal at a site that assigns both the cancel-input and abort-output-block functions to the ASCII

ct AB=ab [cr]

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
ab	Specifies the new abort output block character. This character must differ from the characters defined for the BS, B1, B2, CT, EB, and EL command functions.
[cr]	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-7. Abort-Output-Block Character (AB) Command

character CAN. You want to associate the character CAN only with the concept of cancel-input, so you must redefine the abort-output-block character. The following entries would associate the ASCII character for negative acknowledgment (NAK) with the abort-output-block function:

%AB= (U) [cr]

The CNTRL and U keys generate the NAK code at a 713 terminal. This example assumes that the network control character % is defined by the site for your terminal.

For the remainder of your terminal session, entering the character CAN (using the CNTRL and X keys) cancels an input line, while entering the NAK character aborts an output block in transmission.

REGULATING DEVICE INPUT AND OUTPUT FLOW

The Terminal Interface Program can sometimes send data to terminals much faster than the terminal device can physically process the output. Conversely, if the network is very busy, your terminal can transmit more information than the network can temporarily store before sending the information to the host.

When the first condition occurs, your terminal might use a method called output control to make the TIP stop sending input. When the second condition occurs, the TIP can use a method called input control to make your terminal stop sending input. To use either regulation method successfully, the TIP must know that the terminal can support that method.

You use the IC command to tell the TIP whether your terminal supports device input control. You use the OC command to tell the TIP whether your terminal uses device output control.

The terminal interface program recognizes two characters of input that temporarily suspend and resume input or output. These characters are the ASCII device control 1 (DC1) and device control 3 (DC3) codes.

Another method of implementing flow control which has physical dependencies uses the request to send (RTS) and clear to send (CTS) modem signals. The terminal, or a Local Area Network (LAN), can restrict output by turning off the CTS modem interface signal. CTS output flow control is always recognized and cannot be deselected. The asynchronous TIP can restrict input by turning off the RTS modem signal. This mode of operation is dependent on either special LAN capabilites or the terminal being connected to recognize the RTS signal change as a flow control signal. Because of these dependencies, this method of flow control can be defined only by the Network Administrator. You can use both methods of input flow control simultaneously.

Controlling Output (OC)

If the TIP has been told that your terminal uses output control, the TIP operates as follows. When it receives an ASCII DC3 code from your terminal during output to the terminal, it stops sending data. The DC3 code is not sent to the host as data.

The TIP saves all data from the host until your terminal sends an ASCII DC1 code as the next character of input; the TIP then resumes output with the next stored character.

If your terminal is connected through an X.25 network, the TIP operates as follows. If the TIP has been told that your terminal uses output control, it informs the X.25 network packet assembly/disassembly (PAD) service that your terminal uses output control. When the PAD receives an ASCII DC3 code from your terminal during output to the terminal, it stops sending data. The DC3 code is not sent to the host as data.

The PAD saves all data from the host until your terminal sends an ASCII DC1 code as the next character of input. The PAD then resumes output with the next stored character.

The output control (OC) terminal definition command permits you to state whether your terminal uses this suspend-output method. This command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous terminals). The OC command can also be sent downline to devices in those terminal classes by an application program.

The format of the terminal definition command to change output control use is given in figure 5-8. As shown in tables 5-1 and 5-2, the default choice varies according to terminal class.

```
ct OC [ =option ]  cr

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.

option  Specifies whether your terminal uses
        output control; option can be:

            Y   yes (default if =option is
                omitted)

            N   no

cr      Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-8. Output Control (OC) Command

This control method prevents data loss when using a terminal with buffer that must be cleared periodically before receiving new data. For example, if your terminal is actually a personal computer, it probably uses a buffer to receive output. If your terminal is storing that output as a file on a diskette, it will need to stop the output occasionally so that it can copy the buffer to the diskette.

If your terminal uses the method described earlier, you can make the network software honor it by entering the following command:

```
    %OC=Y  cr
or
    %OC  cr
```

This example assumes that the network control character % is defined by the site for your terminal. After you enter this command, your terminal can suspend output by sending a DC3 and resume output by sending a DC1, for as long as the terminal remains connected to the network (or until the network processing unit is reinitialized).

Controlling Input (IC)

If the TIP has been told that your terminal uses input control, the TIP operates as follows. When it needs to suspend input, it sends an ASCII DC3 code to your terminal during input from the terminal; your terminal then stops sending data. Your terminal saves all data for the host until the TIP sends an ASCII DC1 code; the terminal then resumes input with the next stored character.

If your terminal is connected through an X.25 network, the TIP operates as follows. If the TIP has been told that your terminal uses input control, it informs the X.25 network packet assembly/disassembly (PAD) service that your terminal uses input control. When the PAD needs to suspend input, it sends an ASCII DC3 code to your terminal during input from the terminal; your terminal then stops sending data. Your terminal saves all data for the host until the PAD sends an ASCII DC1 code. The terminal then resumes input with the next stored character.

The input control (IC) terminal definition command permits you to state whether your terminal recognizes this suspend-input method. This command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous terminals). The IC command can also be sent downline for devices in those terminal classes by an application program.

The format of the terminal definition command to change input control use is given in figure 5-9. As shown in tables 5-1 and 5-2, the default choice varies according to terminal class.

```
ct IC [ =option ]  cr

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.

option  Specifies whether your terminal allows
        input control; option can be:

            Y   yes (default if =option is
                omitted)

            N   no

cr      Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-9. Input Control (IC) Command

This control method prevents loss of input data when the data is sent as very large sets of lines or entered by a mechanism other than the keyboard. It can be used with cassette tape readers when the reader starts and stops in response to received codes.

When input control is not used, the network discards lines it cannot store or forward to the host, and sends the message:

```
    WAIT..
```

to your console. When it can again accept input, it sends the message:

```
    REPEAT..
```

to your console. You then must reenter the discarded lines. Reentering the lines is not always convenient, especially when nonconsecutive lines are discarded from a multiple line transmission.

For example, if your terminal is actually a personal computer, you might want to send a file from a diskette as input to the host. If your terminal transmits that file at its maximum speed, the TIP will need to stop the input occasionally until the host acknowledges receipt of what has already been sent.

If the TIP cannot interrupt your terminal's diskette transfer, it just discards portions of that transmission from the middle of the file and you might not be able to easily determine how much of the file reached the host.

If your terminal uses the method described earlier, you can allow the network software to use it by entering the following command:

```
%IC=Y [cr]
```

or

```
%IC [cr]
```

This example assumes that the network control character % is defined by the site for your terminal. After you enter this command, the network will suspend input when it needs to by sending a DC3 and resume input when it can by sending a DC1.

If you have selected input flow control and the device does not stop sending data soon enough after receiving a DC3, and the network must discard input, you receive a DC3 code and the message:

```
WAIT
```

When the network can accept more input, you receive the message:

```
REPEAT..
```

and a DC1 code. To ensure that your file transfers are complete and correct, you should either use a delivery assurance protocol or scan the network output for these messages in the event that some of your input was discarded.

CONTROLLING APPLICATION PROGRAM OPERATIONS

An application program in the host normally processes data from terminals and sends data to terminals much faster than the terminal interface program can physically perform the input and output. Application program operations can sometimes be far ahead of where you think they are because of this lack of synchronization.

If you send many messages to the application program, the program usually processes them in the order sent. In such cases, you might not be able to determine the message currently being processed. The Terminal Interface Program recognizes a terminal definition command to bypass queued input, so that you can send priority data to a host application program. This command is sometimes referred to as a user-interrupt command, although its use does not interrupt any activities significant to you.

The Terminal Interface Program also recognizes two characters of input that do interrupt host application program processing. These characters, called the user-break-1 and user-break-2 characters, cause the TIP to do the following:

Discard all output in progress.

Notify the application program of the interruption.

Identify to the application program the type of break occurring.

Continue to discard application program output until the application program acknowledges receipt of the interrupt information.

You should distinguish between interrupting the Terminal Interface Program and interrupting the host application program. The Terminal Interface Program can be interrupted during output operations by pressing a key sometimes labeled BREAK, ATTN, or INTERRUPT; this interactive or batch output interrupt key is described in section 2. If output is in progress when you want to interrupt the application, you must first interrupt the TIP.

The interactive interrupt key can be defined as performing additional functions. For example, pressing the interactive interrupt key can have the same effect as entering the cancel-input character previously described (see the CN command) or the user-break-1 character (see the B1 command).

Whether the interactive interrupt key is defined as a user-break key does not affect your use of that key to interrupt the TIP. Similarly, you enter the user-break-1 and user-break-2 characters as if they were normal data characters, to be sent to the application program, regardless of whether the interactive interrupt key also performs a user-break-1 or cancel-input function.

The actions the application program takes when it receives the interrupt information depends on the application program, and might differ depending on the type of the break. Traditionally, one input character is used to suspend an application program operation, and another is used to terminate the operation.

For example, the CDC Interactive Facility (IAF) application program interrupts any program operation in process when it receives an indication of a user-break-1 character. IAF sends a message to you indicating this action. IAF ends any program operation in process when it receives an indication of a user-break-2 character. As is done with the user-break-1 character, IAF informs you of the action taken.

You use the break-1 (B1) terminal definition command to change the user-break-1 character, the break-2 (B2) command to change the user-break-2 character, and the break (BR) command to redefine the interpretation of interractive interrupt key use.

Sending a Priority Data Character

The terminal definition command used for this purpose is the only command that has no mnemonic. The command consists of a value only, as shown in figure 5-10.

```
ct a [cr]

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.
        You must press the ATTN key on a
        2741-compatible device before entering
        this character.

a       Specifies any uppercase or lowercase
        alphabetic character recognized by the
        host application program as a
        single-character command.

[cr]    Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-10. Priority Data (User-Interrupt Character) Command

This command is valid from any console device in terminal classes 1 through 18. It cannot be sent downline by an application program.

The valid values depend on the application program with which you are communicating. Any letter in the ASCII character set is allowed by the Terminal Interface Program.

You would normally use this command when you change your mind about prior unprocessed input or need to determine the status of that input. For example, the CDC Interactive Facility (IAF) recognizes several characters for status enquiries. If you begin executing a time-consuming job through IAF, you cannot enter an ENQUIRE command until that job finishes, but you can enter the following at your console:

```
%E [cr]
```

The character % is defined as the network control character for your console. IAF responds to this message by displaying a message similar to the following:

```
JSN: AANY  SYSTEM: BATCH  SRU:  26.344
STATUS: WAITING FOR STORAGE
```

even if your job has not finished execution.

Changing the User-Break-1 Character (B1)

When you enter the user-break-1 character as the only character in a logical line, the terminal interface program sends a break indication of type 1 to the application program currently communicating with the terminal. Any output queued for delivery to the terminal by the TIP is discarded. Output queued by the application program might also be discarded.

The break-1 (B1) terminal definition command permits you to change the character used for this user-break-1 operation. This command is valid from console devices in terminal classes 1 through 15 and 18 (asynchronous, X.25, mode 4, 3270, and HASP terminals). The B1 command can also be sent downline for devices in those terminal classes by an application program.

You must use a character in the 128-character ASCII character set as a user-break-1 character, with the restrictions described earlier in this section. The format of the terminal definition command to change the user-break-1 character is given in figure 5-11. As shown in tables 5-1 and 5-2, the default character varies according to terminal class.

```
ct B1=b1 [cr]

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.
        You must press the ATTN key on a
        2741-compatible device before entering
        this character.

b1      Specifies the new user-break-1
        character. This character must differ
        from the characters currently defined
        for the AB, BS, B2, CN, CT, EB, and EL
        command functions.

[cr]    Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-11. User-Break-1 Character (B1) Command

You would normally change the user-break-1 character when the character used for this function is a character frequently used as the first character in an entry, or when the character used cannot be transmitted from you terminal. For example, suppose you define the : as a tab character when using a text editor, but your console is defined as using that character for the user-break-1 function. During creation of a FORTRAN program source file from a CDC 734, the tab character is frequently the first character you enter on a line. From your viewpoint, the tab character then performs two unrelated functions, depending on whether anything follows it in the entry.

Because you can accidentally enter the tab character as the only character in a line, causing an unintentional application program action, you might prefer to change the less frequently used break

character, rather than the more frequently used tab character. To change the user-break-1 character to the unused ; character, you enter the following:

%B1=; [cr]

This example assumes that the network control character % is defined by the site for your console. After you enter this command, the network interprets the character ; when entered in a line by itself as a user-break-1 indicator.

Changing User-Break-2 Character (B2)

When the user-break-2 character is entered as the only character in a logical line, the terminal interface program sends a break indication of type 2 to the application program currently communicating with the terminal. Any output queued for delivery to the terminal by the TIP is discarded. Output queued by the application program might also be discarded.

The break-2 (B2) terminal definition command permits you to change the character used for this user-break-2 operation. This command is valid from console devices in terminal classes 1 through 15 and 18 (asynchronous, X.25, mode 4, HASP, and 3270 terminals). The B2 command can also be sent downline for devices in those terminal classes by an application program.

You must use a character in the 128-character ASCII character set as a user-break-2 character, with the restrictions described earlier in this section. The format of the terminal definition command to change the user-break-2 character is given in figure 5-12. As shown in tables 5-1 and 5-2, the default character varies according to terminal class.

ct B2=b2 [cr]

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
b2	Specifies the new user-break-2 character. This character must differ from the characters currently defined for the AB, BS, B1, CN, CT, EB, and EL command functions.
[cr]	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-12. User-Break-2 Character (B2) Command

You normally change the user-break-2 character when the character used for this function is frequently used as the first character in an entry, or when the character used cannot be transmitted from your terminal. For example, suppose you define the) as a tab character when using a text editor. During creation of a COBOL program source file from a CDC 734, the tab character is frequently the first character you enter on a line. The tab character performs two unrelated functions, depending on whether anything follows it in the entry.

Because you can accidentally enter the tab character as the only character in a line, causing an unintended application program action, you might prefer to change the less frequently used break character, rather than the more frequently used tab character. To change the user-break-2 character to the unused ; character, you enter the following:

%B2=; [cr]

This example assumes that the network control character % is defined by the site for your console. After you enter this command, the network interprets the character ; when entered in a line by itself as a user-break-2 indicator. Note that this example and the example for the B1 command cannot both be used; the same character cannot be defined as both a user-break-1 and a user-break-2 indicator.

Changing the Functions of the Interactive Interrupt Key (BR)

When you press the interactive interrupt key, the console sends a break indication to the Terminal Interface Program. If the interactive interrupt key is not defined as having additional functions, any output queued for delivery to the console by the TIP is saved and input can be entered.

However, the interactive interrupt key can be defined as performing additional functions. When this is done, if input is in process and the terminal does not use an X.25 network, pressing the interactive interrupt key has the same effect as entering the cancel-input character previously described (see CN command). If output is in process or the console is idle, pressing the interactive interrupt key has the same effect as entering the previously described user-break-1 character (see B1 command).

The break functions (BR) terminal definition command permits you to change how the TIP reacts to use of this key. This command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous and X.25 terminals). The BR command can also be sent downline for devices in those terminal classes by an application program.

The format of the terminal definition command to change the interactive interrupt key functions is given in figure 5-13. As shown in tables 5-1 and 5-2, the default setting is for no additional functions.

```
ct BR [ =option ]  cr

ct       Represents the network control
         character for your console as defined
         by the CT terminal definition command.

option   Determines whether the interactive
         interrupt key has the extra functions
         described in the text; option can be:

             Y   yes (default if =option is
                 omitted)

             N   no

cr       Represents a message termination
         character, code sequence, or event.
         Either the end-of-line or end-of-block
         indicator can be used; refer to the EL
         and EB commands later in this section.
```

Figure 5-13. Break Key Functions (BR) Command

You would normally change the functions defined for the interactive interrupt key when:

- You do not need to distinquish between interrupting output at the console and sending a user-break-1 indication to an application program.
- You do not need to distinquish between interrupting output at the console and discarding input from the console.
- You want a quick way to send either a user-break-1 or a cancel-input indication.

For example, suppose the : is defined as the character for the user-break-1 function and the ; character is defined as the character for the cancel-input function. If you are using a printer mechanism for console output, you can prevent nondata graphic characters from appearing on your listing by redefining the function of the interractive interrupt key.

To add the functions of the user-break-1 character and the cancel-input character to that key, you enter the following:

```
%BR=Y  cr
```

This example assumes that the network control character % is defined by the site for your console. After you enter this command, the network will interpret use of the key when pressed during input as a cancel-input function and when pressed during output (or when the console is idle) as a user-break-1 indicator.

Note that the first function would not be possible if your terminal were using an X.25 network for access to the CDC network.

CONTROLLING INPUT OR OUTPUT FORMAT

The format of terminal console input and output is primarily determined by the terminal hardware and the application program with which the console is communicating. The Terminal Interface Program provides you with commands to make limited modifications to the appearance and speed of input and output.

Through terminal definition commands, you (and the application program) can:

- Insert idle characters (null codes, sometimes called idle fill) to delay output until a mechanical repositioning operation is completed.
- Cause or suppress output paging functions by the TIP.
- Define the physical width of the output mechanism.
- Define the physical length of the output mechanism.
- Cause or suppress cursor repositioning by the TIP at the end of blocks or logical lines.

These commands should be used to describe actual console characteristics to the TIP. Entering a valid command that inaccurately describes the physical capabilities of the console might not cause loss of data but does not always produce desirable results. You should not consider these commands as substitutes for formatting of output by an application program before transmission to the console.

Changing Carriage Return Idle Character Count (CI)

Consoles with mechanically driven output mechanisms sometimes cannot move the output mechanism from the end of one line to the beginning of the next line fast enough to keep up with the data being received. When this happens, the characters that would normally begin the next line are either lost while the output mechanism repositions itself, or else they overstrike portions of the preceding line during the carriage return portion of the repositioning operation.

Terminal support software often inserts a string of nonreproducing characters called idle characters into the output whenever a carriage return operation occurs. These idle characters serve to delay the output of the next line of meaningful data until the repositioning operation is finished. This delay preserves the characters that begin each line.

The Terminal Interface Program also performs this idle character insertion. The character code used for the idle character depends on the terminal class. You can choose the number of characters used through a terminal definition command, or you can instruct the TIP to insert the number appropriate for the terminal class.

The number of idle characters you need to use for the delay depends on the line speed and the output mechanism used. The faster the mechanism, the fewer idle characters needed. The longer the physical line of output possible, the more idle characters needed. The higher the line speed, the more characters needed for delay.

When a nonzero number of idle characters is defined for the console's carriage return operations, the TIP inserts the defined number of characters into the output each time the page width of the console is reached (only for terminal classes 1 through 8), and each time the output contains a character that causes a carriage return operation.

The carriage-return-idle (CI) terminal definition command permits you to change the number of idle characters used during a carriage return operation. This command is valid from console devices in terminal classes 1 through 8 (asynchronous and X.25 terminals). The CI command can also be sent downline for devices in those terminal classes by an application program.

The format of this command is given in figure 5-14. As shown in table 5-1, the default value is zero for devices in terminal classes where the output mechanism is presumed to be a display; display mechanisms require no mechanical repositioning between lines because the beginning of the next line is logically contiguous to the end of the current line.

ct CI=ci [cr]

ct — Represents the network control character for your terminal console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

ci — Specifies the new carriage return idle character count; ci can be:

$0 \leq ci \leq 127$

CA

The value of CA (calculated by TIP) restores the carriage return idle character count to the default value for the terminal class. That value is not necessarily the preset value for your console.

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-14. Carriage Return Idle Character Count (CI) Command

Additional idle characters can be useful, however, at a device where the output mechanism is a display. Many such consoles have an attached device to make a hardcopy of anything placed on the console's screen. These devices are sometimes called slave printers because they only repeat information exchanged with the console and are not known to the network software as separate devices. Slave printers usually are electromechanical and can lose characters during repositioning operations.

For example, CDC 713 terminals are console devices in terminal class 2 and have display output mechanisms. Many 713s are equipped with slave thermal printers. To avoid loss of display data from the thermally produced listing, an idle character count of 7 can be used. To cause such an insertion, you would enter:

%CI=7 [cr]

In this example, the site defines the network control character as % for the terminal (a nonzero page width is presumed). After you make this entry, all output where the TIP expects a carriage return operation to be performed will have seven ASCII null characters (the idle character for terminal class 2) inserted.

These additional idle characters can cause a perceptible pause at the page width boundary of the display or at the beginning of each line of display output. If you find this pause inconvenient when the thermal printer is not being used, the following entry:

%CI=CA [cr]

causes the TIP to revert to the default number of idle characters for terminal class 2, which is zero during carriage return operations.

Changing Line Feed Idle Character Count (LI)

Terminal support software often inserts idle characters during line feed operations for the same purpose as those used during carriage return operations. Console devices with mechanical output mechanisms can smear, lose, or misalign characters of data that would normally be produced after a line feed operation was completed. To avoid such problems, the idle character method of output delay is used.

The Terminal Interface Program also performs this function. The character code used for the idle character depends on the terminal class. You can choose the number of characters used through a terminal definition command, or you can instruct the TIP to insert the number appropriate for the terminal class.

The number of idle characters you need to use for the delay depends on the line speed and the output mechanism used. The faster the mechanism, the fewer idle characters needed. The wider the space between physical lines, the more idle characters needed. The higher the communication line speed, the more characters needed for delay.

When a nonzero number of idle characters is defined for the console's line feed operations, the TIP inserts the defined number of characters into the output each time the page width of the console is reached (only for terminal classes 1 through 8), and each time the output contains a character that causes a line feed operation.

The vertical line feed operation is associated with the page width boundary and a carriage return character because the carriage return operation occurring in both cases is logically distinct from the line feed operation. A carriage return operation on many devices is a horizontal repositioning to the beginning of the current line only. You see a carriage return operation as positioning the output mechanism to the beginning of the next line because the TIP responds to a carriage return character you entered from the console with a line feed character code sent to the console.

The linefeed-idle (LI) terminal definition command permits you to change the number of idle characters used during a line feed operation. This command is valid from console devices in terminal classes 1 through 8 (asynchronous and X.25 terminals). The LI command can also be sent downline for devices in those terminal classes by an application program.

The format of this command is given in figure 5-15. As shown in table 5-1, the default value is 0 or 1 for devices in terminal classes where the output mechanism is presumed to be a display and vertical character positions are logically contiguous. Such terminals require very little delay for repositioning to subsequent lines.

ct LI=li [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

li — Specifies the new line feed idle character count; li can be:

$0 \leq li \leq 127$

CA

The value of CA (calculated by TIP) restores the line feed idle character count to the default value for the terminal class. That value is not necessarily the preset value for your console.

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-15. Line Feed Idle Character Count (LI) Command

Because line feed operations usually occur in conjunction with carriage return operations, it is not often necessary to provide a separate idle character delay for line feeds. However, some consoles can insert additional line feed codes into output as part of a form feeding option. When you use such a device, a separate line feed idle character delay might be necessary.

For example, a device in terminal class 2 usually operates with a line feed idle character count of 0. A terminal emulating a CDC 713 might have a display for an output mechanism but have a slave printer attached for hardcopy. If you use such a terminal with a form feeding option on its printer, additional idle characters might prove necessary.

To provide three additional characters, enter the following:

%LI=3 [cr]

After you make this entry, all line feed operations on the device will include three idle character codes. When you want to discontinue using the form feed option, the extra idle characters can also be removed by entering the following:

%LI=CA [cr]

All subsequent line feed operations will exclude any idle character, because 0 is the default line feed idle character count for terminal class 2.

Changing Page Wait Operation (PG)

Application programs can produce large amounts of output. Such output can be reproduced on your console much faster than you can absorb the information. When no hardcopy device is available at the console, you might have valuable information disappear before you can react to it.

The Terminal Interface Program can perform a function called page waiting that compensates for this situation. You use a terminal definition command to enable or disable the page wait feature.

If your terminal does not scroll its console output, you can use the page wait feature to avoid overwriting information not yet read. If your console scrolls output faster than you can read it, you can use the page wait feature to interrupt scrolling temporarily.

During normalized mode output, the TIP divides each output message into pages on the basis of the page length and page width declared for the console. (Page wait has no effect when a zero page length is defined for the device.) If page wait is supposed to occur, the TIP displays one line less than the declared page length permits and delays further output until terminal operator acknowledgment occurs.

There is one circumstance where output is not delayed indefinitely. This occurs when typeahead is used and the connection is in regulation because of too many unacknowledged input messages. When this occurs, page wait lasts 30 seconds. Output resumes after this delay.

The TIP views the beginning of each downline message as the start of a new page. The TIP calculates the length of a page from the current values of the page width and page length (see the PW and PL commands). If the page width is infinite (PW is zero), a page consists of one line less than the number of logical lines specified as the page length. If the page width is finite (PW is nonzero), a page consists of one line less than the number of physical lines specified by the page length; the TIP cal-

culates the number of physical lines by dividing each logical line into units less than or equal to the defined page width. The last line on the console is always left for terminal operator acknowledgment or other input.

If the TIP detects a top-of-form or clear-screen format effector in the output before the page is filled or output is finished, it also pauses for acknowledgment because processing the format effector would overwrite or destroy data being displayed. In this case, you see the prompting message:

```
OVER..
```

The top-of-form or clear-screen format effector is not processed until you acknowledge receiving the delivered page. The OVER prompt is also used when a double or triple-space format effector occurs at the end of a page.

When page wait occurs, you can enter an empty line as an acknowledgment to the Terminal Interface Program. Empty line acknowledgment entries are not sent upline as input data; any other input acts as an acknowledgment entry and is processed normally. For example, you can use a terminal definition command as both an acknowledgment entry and a functional command; you can enter a line of data as an acknowledgment entry and have it sent upline as normal data.

The paging (PG) terminal definition command permits you to select or suppress page waiting during output operations. This command is valid from console devices in terminal classes 1 through 8, 10 through 13, 15, and 18 (asynchronous, X.25, mode 4, and 3270 terminals). The PG command can also be sent downline for devices in those terminal classes by an application program.

The format of this command is given in figure 5-16. As shown in tables 5-1 and 5-2, the default used is no page waiting for asynchronous and X.25 devices, and page waiting in effect for synchronous devices.

```
ct PG [ =option ]  cr
```

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
option	Specifies whether page waiting should occur; option can be: Y yes (default if =option is omitted) N no
cr	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-16. Page Waiting (PW) Command

You use the PG command to change the page wait status of the console. When a PG value of Y is declared, the console potentially stops at the end of each output page for your acknowledgment before output can continue. Whether page wait actually occurs depends on declaration of a nonzero page length (PL command).

When a PG value of N is used, page waiting is not in effect at the console. In this case, normalized mode output messages are divided by the Terminal Interface Program into full pages, and no pauses for acknowledgment occur during output. If you enter empty lines when page wait is not in effect, those lines are sent upline as empty input messages.

When a PG value of Y is used during transparent mode output, the TIP interprets the end of a message as the end of the page. The page length value declared for the console or embedded character codes performing top-of-form or clear-screen operations do not cause pauses for acknowledgment.

Because the occurrence of a page wait partially depends on the choices in effect for other terminal definition parameters, you should enter the PG parameter at the end of a set of terminal definition command entries. For example, suppose you want to cause page waiting on a CDC 713 that uses % as the network control character. You would enter the following as separate commands:

```
%PL=16  cr
%PG=Y   cr
```

The first command establishes the page length boundary where waiting occurs, and the other enables the option.

The OP command is described later in this section under Selecting an Output Mechanism. The PL and PW commands that define page boundaries are described following this subsection.

The TIP calculates the number of lines output since the last page wait pause, as described for the PL command later in this subsection. If the page width in effect for the console is not the actual page width of the display mechanism, the number of lines constituting a page might not visually correspond to the page length you have selected for page waiting purposes.

Each time the TIP encounters an output line that is longer than the defined page width of the console, the TIP counts that line of output as two or more lines on the assumption that the output mechanism will cause line folding at the page width in effect. The value in effect for the page width therefore indirectly affects the apparent page boundaries.

In the example just given, a page wait for user acknowledgment occurs every 15 lines of all subsequent normalized mode output until you enter a terminal definition command that turns off page waiting. You can explicitly disable the function by entering the following:

```
%PG=N  cr
```

If you enter the following instead:

```
%PL=0  cr
```

page wait pauses no longer occur, and no extra vertical space is inserted; the TIP cannot calculate the boundary of an infinitely long page.

Changing Page Width (PW)

Many of the operations performed by the Terminal Interface Program depend on the way the console's output mechanism is defined. One of the more important definitions is the number of characters the output mechanism can contain as a single line.

This maximum number establishes the page width boundary for all normalized mode output. The page width of the console is always defined in terms of maximum number of output characters per line, because the maximum number of input characters per line depends on where the previous line of output left the cursor or type element positioned.

You might see the effect on output of an inappropriate page width definition as the generation of several lines where one is expected, or as a pause for page wait in what looks like the wrong place. The effect on output depends on the output mechanism defined for the console. Some of the effects on output are described earlier in this section under Changing Page Wait. The remaining effects are described in the following paragraphs.

The Terminal Interface Program uses the defined page width of the console to determine when lines should be folded during output to printer mechanisms. The TIP also uses the page width to determine when display mechanisms will perform automatic line folding, but require idle character insertion.

If your console is defined as having a printer output mechanism and the current page width definition is less than or equal to the physical page width of the console, the Terminal Interface Program causes a carriage return and line feed operation after each occurrence in the output line of the defined number of characters. If the defined page width of these consoles is greater than the physical page width, characters usually are either truncated or overstruck on the right margin of the output device.

If your terminal has a display as an output mechanism and the page width definition is less than the physical page width of the output device, the TIP might seem to pause at the page width boundary. This pause occurs when it inserts any required idle characters in anticipation of a carriage return and line feed operation at the incorrect page boundary. In this case, the TIP does not insert idle characters at the real page boundary.

If the defined page width is greater than the actual page width, characters:

Might be truncated

Might be overstruck on the right margin of the output device

Might replace characters at the left margin of the same line

Might be continued on the next line

Most terminals with display mechanisms for output automatically continue the output line on the next line. When this automatic line folding occurs and the defined page width is different than the actual page width, the TIP does not insert idle characters at the proper place in the output.

A nonzero page width definition is not meaningful for devices with a paper tape punch or paper tape emulator as the defined output mechanism. You can use a zero (infinite) page width safely for such devices. If your terminal simultaneously prints output during explicit paper tape output operations, you might need to use a nonzero page width so that the TIP will provide the appropriate carriage return, line feed, or idle characters needed for visual fidelity of the data.

The page width (PW) terminal definition command permits you to change an inappropriate page width value. This command is valid from console devices in terminal classes 1 through 18. The PW command can also be sent downline by an application program.

The format of this command is given in figure 5-17. As shown in tables 5-1 and 5-2, the default value used varies according to terminal class, but is always nonzero.

ct PW=pw [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

pw — Specifies the page width in characters; pw can be:

- 0 — Page is infinitely wide (output line length is unimportant; input lines are presumed to be 100 characters or less)
- $20 \leq pw \leq 255$

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-17. Page Width (PW) Command

There is a functional restriction on the page width value used for some mode 4A terminal consoles. These terminals require the product of the page width and page length values used to be less than or equal to the transmission block size. For example, the page width you choose for a CDC 731-12 (terminal class 15) must be such that the total number of characters on a page is less than or equal to 1039. When the page length defined for this terminal device is 13 lines, the page width must be 80 characters or less to meet the transmission block size restriction.

An additional function restriction might exist for all consoles of any terminal class. The application program currently servicing the console might use the currently defined page width value to format output. The Terminal Verification Facility

described in section 7 is an example of such a program. A zero page width might produce undesirable results while communication with such programs exists. In general, you should not use a zero page width for devices in terminal classes 9 through 17. You cannot use a zero page width for terminal class 18.

Inappropriate page width definitions normally occur when the physical device using a communication line is not the device for which the corresponding logical terminal device was configured. Automatic recognition lines cause the most frequent occurrence of this situation. This lack of correspondence also occurs when terminal consoles with different display formatting options installed use the same line.

For example, console devices of terminal classes 10 and 11 are available with line width options of either 50 or 80 characters. A terminal with the 50-character option installed can use a line configured for terminals with the 80-character option, or vice versa. You can avoid any confusion by entering the following from a 50-character terminal when connection to the network is completed:

```
%PW=50 [cr]
```

Changing Page Length (PL)

The output operations performed by the terminal interface program also depend on the page length defined for the output device. The page length defined for the terminal establishes the vertical page boundary for the terminal output device. Depending on the device used, the page length value is either the maximum number of lines that can appear simultaneously on the device, or the maximum number of output lines that will fit within the vertical boundaries of a form.

The page length of the console is always defined in terms of maximum number of output lines of the defined page width. Page length is not significant to you during input operations. The Terminal Interface Program does not try to correlate output line position on the page with input line position on the page.

When a console has a nonzero page length, each line of normalized mode output completed causes the TIP to increment a line counter. The line counter is reset to zero each time the page boundary is reached, each time a top-of-form or clear-screen code is encountered in the output, and each time a change in the direction of message traffic takes place.

The latter condition occurs regardless of whether the TIP completes all output to the console and expects input from it. Input from consoles that can interrupt output, or input simultaneously during output, always resets the line counter. Changing from one nonzero page length to another also resets the line counter to zero.

When the console has a zero page length, the TIP does not increment a line counter or perform any functions dependent on page length. A zero page length is equivalent to an infinite length. The line counter remains at zero until a nonzero page length is defined and a line of output is encountered.

You usually see the effect on output of an inappropriate page length definition as the gradual scrolling of the page boundary. On some mode 4A terminals, such as the CDC 731-12, there is a functional restriction on the page length value used. These terminals require the product of the page length and page width values used to be less than or equal to the transmission block size of the terminal. If your terminal belongs in class 15, the page length you choose should be such that the total number of characters on a page is less than or equal to 1039. When the page width defined for such a console is 80 characters, the page length must be 13 lines or less to meet the transmission block size restriction. If you do not meet this restriction, output transmission problems can occur.

An additional functional restriction might exist for all consoles in any terminal class. The application program currently servicing the console might use the currently defined page length value to calculate transmission block size or otherwise format output.

The Terminal Verification Facility described in section 7 is an example of such a program. A zero page length might produce undesirable results while communication with such programs exists. In general, you should not use a zero page length for terminals in classes 10 through 17. You cannot specify a zero page length for terminal class 18.

The Terminal Interface Program uses the defined page length of the console to determine when page spacing or page wait operations should be performed. The operation performed by the TIP when the page length boundary is reached depends on the terminal class and the page wait selection in effect. The effects of the page length value defined on use of the page wait feature are described earlier in this section under the heading Changing Page Wait.

When the TIP performs page waiting, the effective page boundary is one line less than the defined page length. When this effective page boundary is reached, the TIP returns the cursor to the left margin of the following line and waits for you to acknowledge the page.

Acknowledgment entries cause a carriage return and line feed operation. Also, subsequent output is frequently produced after a line feed operation.

Because of this spacing, the page length you choose for a nonscrolling console during page wait operations can result in slow scrolling of the page boundary, even when a value appropriate for the terminal is used. You can correct such scrolling by selecting a page length one line shorter than normal during page wait operations, then changing the defined length back to its maximum value when page waiting is disabled.

A nonzero page length is not meaningful for terminals with a paper tape punch or paper tape emulator defined as the output mechanism. You can use a zero (infinite) page length safely at such terminals.

The page length (PL) terminal definition command permits you to change an inappropriate page length value. This command is valid from console devices in terminal classes 1 through 18. The PL command can also be sent downline by an application program.

The format of this command is given in figure 5-18. As shown in tables 5-1 and 5-2, the default value used is nonzero only for consoles in terminal classes where the output mechanism is a display by default.

ct PL=pl [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

pl — Specifies the page length in physical lines; pl can be:

0 Page is infinitely long

$8 \leq pl \leq 255$

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-18. Page Length (PL) Command

The values chosen for the defaults of mode 4 and 3270 terminals (terminal classes 10 through 13, 15, and 18) were selected according to the most commonly installed display options for these terminals. Table 5-5 lists all page length and page width options offered by CDC for terminals in these classes.

TABLE 5-5. MODE 4 TERMINAL PAGE WIDTHS AND LENGTHS

Terminals	Terminal Class	Option 1 (Default)		Option 2	
		Length	Width	Length	Width
200 User Terminal	10	13	80	20	50
714-30/40	11	16	80	24	50
711-10	12	16	80	8	80
714-10/20	13	16	80	8	80
731-12/ 732-12	15	13	80	N/A	N/A
734	15	13	80	N/A	N/A

Note: N/A means not applicable.

Inappropriate page length definitions normally occur when the physical terminal using a communication line is not the terminal for which the corresponding logical terminal was configured. Automatic recognition lines cause the most frequent occurrence of this situation. This lack of correspondence also occurs when terminal consoles with different display formatting options installed use the same line.

For example, a console of terminal class 10 with a 20-line option installed can use a line configured for terminals with the 13-line option consoles, or vice versa. You can avoid any confusion by entering the following from a 20-line terminal console when connection to the network is completed:

%PL=20 [cr]

%PW=50 [cr]

Controlling Cursor Positioning (CP)

The Terminal Interface Program normally responds to input transmission delimiters by repositioning the console's cursor or printing mechanism to the beginning of the next physical line. It sends a carriage return code in response to a line feed from the keyboard (IN command KB value), and can send a carriage return and/or a line feed code as a response to an end-of-line or end-of-block code. The response used for cursor positioning is determined by the cpr value of the EL or EB command, described later in this section. You can control whether the TIP performs cursor positioning with the cursor-positioning (CP) terminal definition command.

The CP command has the format shown in figure 5-19. This command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous and X.25 terminals). The CP command can also be sent downline to devices in those terminal classes by an application program.

ct CP [=option] [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command.

option — Indicates whether to send a response to the console; option can be:

Y yes (default if =option is omitted)

N no

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-19. Cursor Positioning Response (CP) Command

The default values used for the cursor positioning selection are listed in tables 5-1 and 5-2. The range of values possible for each terminal class are shown in tables 5-3 and 5-4.

If CP=Y and IN=BK, the TIP does not respond to line feeds or to end-of-line codes. Only the response defined by the EB command is used.

You normally suppress cursor positioning when your console performs it automatically. For example, suppose you were using a CDC 721 Viking Extended terminal that scrolls its output, defined with % as its network control character. Your terminal was set up so that it performs a carriage return and line feed when the CR key is pressed. You can prevent double-spacing of multiple input lines by typing the following command before you receive the output you need to copy:

```
%CP=N [cr]
```

Cursor postioning stops after the next output to your console. Thereafter, each of your entries repositions the cursor to the beginning of the next physical line only.

CONTROLLING INPUT OR OUTPUT TRANSMISSIONS

During input or output transmissions, the terminal interface program performs operations on the transmitted data. These operations convert data from a form intelligible to the source of the data to a form intelligible to the destination of the data.

The TIP performs the following operations:

- Interprets, sets, and clears parity bits in terminal code character bytes
- Echoplexes characters (repeating each character of input as output to confirm correct receipt of the character)
- Determines communication line speed for asynchronous terminals using certain lines
- Translates terminal character set codes to and from ASCII character codes
- Edits character input before transmission to the host application program
- Processes terminal definition commands and all function flag characters (performs actions for user-break-1, user-break-2, and so forth)
- Inserts and removes transmission protocol information that is dependent on the input or output device used
- Edits normalized output before transmission to the terminal (removes unit separators and processes format effectors, as described in section 3)

General TIP operations, and many of these functions, are described in section 2.

The terminal interface program supports two major transmission modes, called normalized mode and transparent mode. When a terminal operates in normalized mode, the TIP performs most or all of the functions listed. Normalized mode allows three alternatives for input editing:

- Full editing
- Special-editing
- Full-ASCII mode

These modes are described in section 2.

When a terminal operates in transparent mode, the TIP does not perform the following functions:

- Edit character input or output
- Translate to and from ASCII character codes
- Process terminal definition commands or any function flag characters

All other TIP operations are performed in transparent mode.

Input transmission mode is separate from output transmission mode, just as the input mechanism used is separate from the output mechanism used. You select transparent input transmission mode through a terminal definition command.

Termination of transparent input transmission mode occurs when the TIP detects a condition that you or the network application program define to it in advance. Transparent input transmission mode also ends when your console is disconnected from a host.

The initial and default input transmission mode is normalized mode. Selection of output transmission mode and termination of it are controlled by the application program with which the console communicates.

The Terminal Interface Program performs character translation to and from parity-free ASCII character codes for consoles using the following terminal character set codes:

- ASCII code with parity
- ASCII code with parity for APL bit-pairing character sets
- ASCII code with parity for APL typewriter-pairing character sets
- CDC external BCD with odd parity
- IBM correspondence code for manifold character set with odd parity
- IBM correspondence code for APL character set (print element 1167987) with odd parity
- IBM EBCD for EBCD graphic character set with odd parity
- IBM EBCD for APL character set (print element 1167988) with odd parity
- IBM EBCDIC for PN print train character set

The automatic recognition process described in section 4 allows the TIP to correctly identify the communication line speed and terminal code set without your knowing it. The logical terminal can also be configured so that both the line speed and the character set in use are known when connection to the network occurs.

The APL character set is the only character set supported that is generally associated with more than one code set. The APL character set is also the only set that you normally switch to or from during a terminal session.

You use a terminal definition command to make the TIP aware that a change to or from APL character set use is occurring at the console. This command allows the TIP to compensate in its code translations for the change in character-to-code conversion that occurs at the console.

Some application programs use two of the characters the TIP normally recognizes as function flag characters and removes from the input. A terminal definition command allows you to generate these characters as data without causing TIP editing or other functions. This special editing command provides suppression of portions of the normal editing of normalized mode input data.

Changing Line Speed or Character Set Use (AR)

When you need to physically change the line speed or the character set associated with the terminal input and output devices, you must inform the TIP of the change. You use the automatic recognition (AR) terminal definition command to do this.

This command has the format shown in figure 5-20. It is valid from console devices in terminal classes 1 through 8 (asynchronous terminals). The AR command cannot be sent downline by an application program.

ct AR [cr] CR) CR

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

CR — Represents the key that generates a carriage return code.

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-20. Automatic Recognition (AR) Command

If your terminal is connected through an X.25 network, only the character set associated with your terminal can be changed with the AR command. In order for the TIP to correctly recognize the character set associated with your terminal, the X.25 network packet assembly/disassembly (PAD) service must be able to pass untranslated characters to the TIP.

As shown in tables 5-1 and 5-2, the default character set assumed for code translation is never an APL set. As indicated in tables 5-3 and 5-4, the possible alternative sets are partially dependent on terminal class. These alternatives are identified by numbers, as follows:

Set Number	Character Code Set
1	CDC External BCD
2	ASCII
3	ASCII typewriter-pairing APL
4	ASCII bit-pairing APL
5	IBM Extended BCD
6	IBM Extended BCD APL
7	Correspondence code
8	Correspondence code APL
9	EBCDIC

As shown in figure 5-20, the method used to change the code set translation does not require you to be aware of the sets actually used by the terminal. You enter the command in two parts, between which you make the actual changes at the console.

For example, to change a Diablo 1641 console (terminal class 1) from its normal ASCII code set to an ASCII APL code set, you perform the following procedure. First, you enter the command:

%AR [cr]

The site defines the network control character as % for your console. You then:

Change the switch that selects the console line speed (omit this step if you know you are not allowed to change line speed).

Change the switch that selects the modem speed on the attached modem (omit this step if you know you are not allowed to change line speed).

Transmit a carriage return code to the network.

Wait for two line feeds as a response; if you are not allowed to change your line speed, no line feed responses occur and your subsequent entries cannot be recognized until you change your console and modem speed to its original value.

Change the switch that selects the console keyboard set.

Change the character daisy wheel to the appropriate APL set.

Enter:

)CR

Wait for a single line feed response

When the TIP acknowledges the last entry with an

AR ACCEPTED..

message, the code translation change matching the console character set change is completed.

Note that this procedure is similar to the terminal identification procedure performed when an asynchronous terminal originally establishes access to the network. The same functions are performed by both procedures.

After you have finished using the APL interpreter, you might want to change the character code set back to the console's normal set so that you can visually check commands entered to other software. To change the character code set back, you perform the same two-step procedure.

First, you enter the command:

÷ AR [cr]

The network control character ÷ in this entry is the APL character set equivalent of the normal character set % symbol, which is defined by the site as the control character for your console. You then enter a carriage return, change the switch that selects the console code set, and enter:

)CR

Wait for a single line feed response

In these examples, the order in which you perform the steps is significant. You can change the order from the sequence of enter command, change set, type parenthesis command. Changing the order invalidates the example by changing the console characters involved. For example, if you make both entries before you make the physical change to the APL set, your second entry must be:

"CR

Wait for a single line feed response

because the character) in the APL character set is equivalent to the character " in the ASCII character set.

Using Special Editing (SE)

During most normalized mode input operations, the Terminal Interface Program removes character codes from the input and performs appropriate functions for them. This editing function provides you and the application program with consistent operational responses and reduces the amount of nondata traffic through the network.

Sometimes, characters involved in TIP input editing functions are valid data that need to be transmitted to the network application program. The characters for which this is most frequently true are the backspace character code and the line feed character code (described earlier in this section).

You can bypass normal TIP editing operations for these characters by selecting the special editing option for normalized mode input. You select or reject this option by entering the special editing (SE) terminal definition command.

This command has the format shown in figure 5-21. This command is valid from console devices in terminal classes 1 through 8 (asynchronous terminals). The SE command can also be sent downline for devices in these terminal classes by an application program.

ct SE [=option] [cr]

ct	Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.
option	Specifies whether to use the special-editing input mode; option can be:
	Y yes (default if =option is omitted)
	N no
[cr]	Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-21. Special-Editing Mode (SE) Command

As indicated in tables 5-1 and 5-2, the default normalized mode editing option allows full TIP editing of input. The effects of selection of the special editing option during normalized mode input are shown in table 5-6.

Special editing is only performed when full-ASCII editing has not been selected and transparent mode input is not occurring. Refer to the FA and IN commands. When the special editing option is in effect, the exempted characters are transmitted to the host as data.

For example, if you enter a backspace character from a Texas Instruments Silent 700 console (terminal class 1) while full editing is not in effect, a backspace operation occurs at the terminal, the TIP discards the backspace character code and deletes the previously entered character from the NPU input buffer. This occurs whether the ASCII or ASCII-APL character set is in use at the terminal.

Suppose you enter the following command:

%SE=Y [cr]

The site defines the network control character as % for your console. After you make this entry, any backspace character you enter causes a backspace operation at the console, does not delete a char-

acter from the TIP input buffer, and stores the backspace character code for transmission to the host.

Input editing exemption occurs for the line feed character code. The line feed character code is stored for transmission, a normal line feed operation occurs at the terminal, and the input line stored is not transmitted to the host.

An additional operation occurs if you enter a backspace followed by a line feed; the TIP sends a bell code to your console.

Using Full-ASCII Editing (FA)

During most normalized mode input operations, the Terminal Interface Program removes many nongraphic character codes from the input and performs appropriate functions for some of them. This editing function provides you and the application program with consistent operational responses and reduces the amount of nondata traffic through the network.

Sometimes, characters discarded by TIP input editing functions are valid data that need to be transmitted to the network application program. These characters include all the ASCII control characters generated by the console (see appendix A). These character codes are meaningful data to some software.

You can bypass normal TIP editing operations for these characters by selecting the full-ASCII option for normalized mode input. You select or reject this option by entering the full-ASCII (FA) terminal definition command. This command has the format shown in figure 5-22.

```
ct FA [ =option ]  cr

ct      Represents the network control
        character for your console as defined
        by the CT terminal definition command.
        You must press the ATTN key on a
        2741-compatible device before entering
        this character.

option  Specifies whether to use the full ASCII
        input editing mode; option can be:

            Y   yes (default if =option is
                omitted)

            N   no

cr      Represents a message termination
        character, code sequence, or event.
        Either the end-of-line or end-of-block
        indicator can be used; refer to the EL
        and EB commands later in this section.
```

Figure 5-22. Full-ASCII Editing Mode (FA) Command

This command is valid from console devices in terminal classes 1 through 8, 10 through 13, and 15 (asynchronous, X.25, and mode 4 terminals). The FA command can also be sent downline for devices in these terminal classes by an application program.

As indicated in tables 5-1 and 5-2, the default normalized mode editing option allows full TIP editing of input. The effects of selection of the full-ASCII editing option during normalized mode input are shown in table 5-6.

Full-ASCII editing is only performed when transparent input is not occurring. Special editing is only performed when full-ASCII editing has not been selected and transparent mode input is not occurring. Refer to the SE and IN commands.

When the full-ASCII editing option is in effect, the exempted characters are transmitted to the host. The functions associated with the characters defined by the AB, CN, B1, B2, and BS commands and the line feed code are not performed.

Those character codes are stored in the data. Although the end-of-line and end-of-block indicators retain their functions, those codes are also stored.

For example, if you enter one of the following ASCII characters from a console while full editing or special editing is not in effect:

NUL, DEL, STX, LF, CR, EOT, ETX

DC3 or DC1 when output control is in effect (OC is Y)

the TIP discards the code. This occurs regardless of the character set in use by the console. If the terminal generates codes that are converted to one of these ASCII codes, the conversion still occurs.

Suppose you enter the following command:

%FA=Y cr

The site defines the network control character as % for your console. After you make this entry, you can enter any of the control characters above and store the corresponding ASCII code for transmission to the host.

The only exceptions to this storage are the following ASCII codes:

NULs, DELs, and LFs when encountered as the first character of an input line or as the character following the end of a logical line or end of a block

CRs that follow LFs when the TIP knows that input is occurring from paper tape (IN is PT)

DC3 or DC1 when output control is in effect (OC is Y)

DC3s that follow the end of a logical line when the TIP knows that input is occurring from paper tape (IN is PT)

TABLE 5-6. INPUT CHARACTER EDITING SUMMARY

Editing Option in Effect For Transmission	ASCII Characters Removed From Input Before Transmission To Host When Occurring at						
	Beginning of Message Before First Character Stored as Data	First Data Character Followed by End of Message (No Page Wait)	First Data Character Followed by End of Message (Page Wait)	Middle of Message (Anywhere)	Middle of Message (After Line Feed)	Preceding End of Message	End of Message
	FULL EDITING						
SE=N, FA=N, OC=N, IN=KB	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) Line feed (LF) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	Null (NUL) Delete (DEL) Backspace (1) End-of-message (2)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	End-of-message (2) (4)
SE=N, FA=N, OC=N, IN=PT	Null (NUL) Delete (DEL) Backspace X-OFF (DC3) Line feed (LF)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) Line feed (LF) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	Null (NUL) Delete (DEL) Backspace (1) End-of-message (2)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	End-of-message (2) (4)
SE=N, FA=N, OC=N, IN=BK	Null (NUL) Delete (DEL) Backspace Beginning-of-message (3) Line feed (LF)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	Null (NUL) Delete (DEL) Backspace (1) End-of-message (2)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF)	End-of-message (2) (4)
SE=N, FA=N, OC=Y, IN=KB IN=PT	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Network control character (1) Line feed (LF) User-break- (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) Backspace (1) X-OFF (DC3) X-ON (DC1) End-of-message (2)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF) X-OFF (DC3) X-ON (DC1)	End-of-message (2) (4)
SE=N, FA=N, OC=Y, IN=BK	Null (NUL) Delete (DEL) Backspace (1) Beginning-of-message (3) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Network control character (1) Backspace (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) Backspace (1) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) Backspace (1) X-OFF (DC3) X-ON (DC1) End-of-message (2)	Null (NUL) Delete (DEL) Backspace Line feed (LF) X-OFF (DC3) X-ON (DC1)	End-of-message (2) (4)

TABLE 5-6. INPUT CHARACTER EDITING SUMMARY (Contd)

Editing Option in Effect For Transmission	ASCII Characters Removed From Input Before Transmission To Host When Occurring at						
	Beginning of Message Before First Character Stored as Data	First Data Character Followed by End of Message (No Page Wait)	First Data Character Followed by End of Message (Page Wait)	Middle of Message (Anywhere)	Middle of Message (After Line Feed)	Preceding End of Message	End of Message
PARTIAL EDITING							
SE=Y, FA=N, OC=N, IN=KB	Null (NUL) Delete (DEL)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) End-of-message (2)	Null (NUL) Delete (DEL)	End-of-message (2) (4)
SE=Y, FA=N, OC=N, IN=PT	Null (NUL) Delete (DEL) X-OFF (DC3)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) End-of-message (2)	Null (NUL) Delete (DEL)	End-of-message (2) (4)
SE=Y, FA=N, OC=N, IN=BK	Null (NUL) Delete (DEL) Beginning-of-message (3)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) End-of-message (2)	Null (NUL) Delete (DEL)	End-of-message (2) (4)
SE=Y, FA=N, OC=Y, IN=KB IN=PT	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1) End-of-message (2)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	End-of-message (2) (4)
SE=Y, FA=N, OC=Y, IN=BK	Null (NUL) Delete (DEL) Beginning-of-message (3) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break-1 (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Network control character (1) User-break (1) User-break-2 (1) Abort-output-block (1) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1) End-of-message (2)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	End-of-message (2) (4)
FA=Y, OC=N, IN=KB	Null (NUL) Delete (DEL) Line feed (LF)	Network control character (1)	Network control character (1) Line feed (LF)	Null (NUL) Delete (DEL) Line feed (LF)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) Line feed (LF)	
FA=Y, OC=N, IN=PT	Null (NUL) Delete (DEL) Line feed (LF) X-OFF (DC3)	Network control character (1)	Network control character (1) Line feed (LF)	Null (NUL) Delete (DEL) Line feed (LF)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) Line feed (LF)	

TABLE 5-6. INPUT CHARACTER EDITING SUMMARY (Contd)

Editing Option in Effect For Transmission	ASCII Characters Removed From Input Before Transmission To Host When Occurring at: Beginning of Message Before First Character Stored as Data	First Data Character Followed by End of Message (No Page Wait)	First Data Character Followed by End of Message (Page Wait)	Middle of Message (Anywhere)	Middle of Message (After Line Feed)	Preceding End of Message	End of Message
PARTIAL EDITING (Contd)							
FA=Y, OC=N, IN=BK	Null (NUL) Delete (DEL) Line feed (LF)	Network control character (1)	Network control character (1)	Null (NUL) Delete (DEL) Line feed (LF)	Null (NUL) Delete (DEL)	Null (NUL) Delete (DEL) Line feed (LF)	
FA=Y, OC=Y, IN=KB, IN=PT, IN=BK	Null (NUL) Delete (DEL) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Network control character (1) X-OFF (DC3) X-ON (DC1)	Network control character (1) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) Line feed (LF) X-OFF (DC3) X-ON (DC1)	Null (NUL) Delete (DEL) X-OFF (DC3) X-ON (DC1)	
NO EDITING							
SE=N, SE=Y, FA=N, FA=Y, IN=XK, IN=XP, IN=X	Delimiter (4) End-of-message (4) (empty message)	None	None	None	None	None	Delimiter (4) End-of-message (4)

(1) Whichever character is currently defined; this character can be changed by TIP command.

(2) The character code discarded depends on the code used in the communication line protocol for the terminal class; these codes include: end-of-text (ETX), end-of-transmission (EOT), and carriage return (CR).

(3) The character code discarded depends on the code used in the communication protocol for the terminal class; these codes include: start-of-text (STX), start-of-header (SOH), and synchronize (SYN).

(4) The character codes discarded depend on the codes defined as the end-of-message indicator and the transparent mode operation delimiter for the console device.

CONTROLLING END-OF-INPUT INDICATORS AND DELIMITERS

As described in section 2, all console data moves through the network in data blocks. The network forms data blocks from input when a predefined condition is encountered. You can control this block creation by using the BF terminal definition command to change the blocking factor the network uses.

You can input and receive data that is transmitted in either normalized mode or transparent mode. The application program communicating with the console can select either mode for transmission of output. You or the application program can select either mode for transmission of input; you use the IN terminal definition command (described later in this section) for this selection.

There are two alternatives for normalized mode input:

Line mode operation

Block mode operation

Two terminal definition commands allow you to define the normalized mode end-of-input indicators and the Terminal Interface Program's response to them. The end-of-line (EL) command describes the conditions existing when you end a logical line of normalized input; the end-of-block (EB) command describes the conditions existing when you end a logical block of normalized input.

There are two alternatives for transparent mode input:

Single-message input

Multiple-message input

When you use single-message input, ending transparent mode input also ends a single message to the application program. When you use multiple-message input, the end of a message is distinct from the end of transparent mode input. In multiple-message transparent input mode, more than one message can be sent to the application program without requiring you to reenter transparent mode between messages.

Although you can use a terminal definition command to begin transparent mode transmission, you cannot use the same method to end transparent mode messages or transparent mode transmission. Because of this, you use one of two separate commands to define an event to the TIP that identifies the end of a message or the end of transparent mode input operation. The delimiter (DL) command is used to set up your device for single-message input; the extended delimiter (XL) command is used to set up your device for multiple-message input.

Changing the Blocking Factor (BF)

The network software sends a network data block to the host when one of the following conditions occurs:

A logical line ends.

A physical line ends with a line feed and the network has been told that this has significance to the host.

A multiple of 100 characters is received from the terminal.

You can change the blocking factor used by entering the blocking factor (BF) terminal definition command. This command has the format shown in figure 5-23.

ct BF=bf [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

bf — Specifies the blocking factor the network software should use in creating upline network data blocks from your terminal's input; bf can be:

- 0 — Create a block each time a line feed is entered or each time 100 characters are entered
- 1 — Create a block each time 100 characters are entered
- 2 — Create a block each time 200 characters are entered

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands later in this section.

Figure 5-23. Blocking Factor (BF) Command

The BF command is valid from console devices in terminal classes 1 through 8 (asynchronous and X.25). The BF command can also be sent downline for devices in these terminal classes by an application program.

As indicated in tables 5-1 and 5-2, the default blocking factor for all terminal classes is 100 characters per network block, with no special significance attached to input of a line feed. The range of values permitted for the command is given in tables 5-3 and 5-4.

You would normally change the blocking factor when the occurrence of a line feed in your input has significance to the host application program. For example, the Message Control System (MCS) distinguishes between blocks transmitted after a line feed and blocks transmitted after a carriage return. The default blocking factor does not allow MCS to distinguish blocks transmitted after a line feed because entering a line feed does not cause a block to be transmitted.

To use MCS properly, you would enter:

%BF=0 [cr]

The site defines the network control character as % for your console. After you make this entry, MCS is able to distinguish between the two types of blocks.

Defining End-of-Line Processing (EL)

You end a logical line of input by pressing a key that sends a code, code sequence, or event indicator to the Terminal Interface Program. This code, sequence, or event is called an end-of-line indicator in this manual. The TIP can send your console a response to this end-of-line indicator, producing appropriate cursor positioning at the terminal for the next input or output.

You can control both the end-of-line indicator used and the response used by the TIP; whether the response is sent depends on the value used for the CP command, described earlier in this section. You control the indicator and response through the end-of-line (EL) terminal definition command. This command has the format shown in figure 5-24.

The EL command is valid from console devices in terminal classes 1 through 3, 5 through 8, 10 through 13, and 15 (asynchronous, X.25, and mode 4 terminals). The EL command can also be sent down-line to devices in these terminal classes by an application program. The cpr parameter cannot be used for mode 4 devices; the char value cannot be used for mode 4 terminal devices (terminal classes 10 through 13 and 15).

As indicated in tables 5-1 and 5-2, the default end-of-line indicator and cursor positioning response vary acccording to the terminal class. The range of values permitted for each parameter is given in tables 5-3 and 5-4.

You would normally change the end-of-line indicator used when the current value needs to be stored as data, when you want the current end-of-block indicator to be the end-of-line indicator, or when you want to stop that use of the end-of-block indicator (see the EB command in the next subsection). You would change the cursor positioning response when the device itself provides part or all of the necessary repositioning, or when receiving a response would be inconvenient or confusing.

For example, suppose you are using the tty asynchronous teletypewriter emulation option of a Wang Model 30 Word Processing System, defined to the network as terminal class 2. The network control

ct EL= { el / el, cpr / cpr } [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command.

el — Specifies the new end-of-line character; el can be:

el	Description
char	Defines a new character for the end-of-line indicator function. This character must differ from the characters currently defined for the AB, BS, B1, B2, CN, and CT command functions.
EB	Selects the character currently defined as the end-of-block indicator.
EL	Selects the default end-of-line character or character sequence for the terminal class.

cpr — Specifies the cursor-positioning response to the end-of-line indicator; cpr can be:

cpr	Description
CR	Send a carriage return to the console.
LF	Send a line feed to the console.
CL	Send a carriage return and a line feed to the console.
NO	Send no response; this has an effect similar to declaring a value of N for the CP command, except that the response to an end-of-block indicator is unaffected by this choice.

The parameters el and cpr are optional and order-independent. If you omit one, the network retains the previous value for the parameter.

[cr] — Represents a message termination character, code sequence, or event. Either the old end-of-line or end-of-block indicator can be used.

Figure 5-24. End-of-Line Delimiter Control (EL) Command

character defined for your device is %. When transferring a text document from the Wang system disk to a CDC CYBER 170, the Wang transmits a carriage return at the end of each text line; the CYBER receives each text line as a separate message; enroute, the TIP discards the carriage return codes.

You could change the end-of-line indicator to an ASCII formfeed code (transmitted by the Wang at the end of each document page); that action would preserve the carriage return codes. You could also define the cursor response for the end-of-line indicator to be "none"; that action would avoid a cursor response at the end of each page. To do this, you could enter the following command before the text document transfer begins:

```
%EL= ⊥,NO  [cr]
```

The symbol ⊥ represents the Wang page break code. The page break cannot be entered using the PAGE key during teletypewriter emulation, so this command could have been previously stored in another document and transmitted from that document. (L) could also be used.

Response suppression occurs immediately. The next logical line entered must end with a formfeed code to be recognized as a completed line.

You would enter the following command to resume normal operation after the text document is transferred:

```
%EL=EL,YES ⊥
```

Again, you could send this command from a previously stored document because you cannot enter the page break using the PAGE key during teletypewriter emulation. When the TIP receives this command, it changes your terminal's end-of-line indicator back to the default value (indicated by the EL parameter) for terminal class 2, and resumes cursor positioning responses. You must use the default end-of-line indicator, the ASCII carriage return code, to end subsequent messages.

Defining End-of-Block Processing (EB)

You end a logical block of input by pressing a key that sends a code, code sequence, or event indicator to the Terminal Interface Program. This code, sequence, or event is called an end-of-block indicator in this manual. The TIP can send your console a response to this end-of-block indicator, producing correct cursor positioning for the next input or output. When your terminal is operating in block mode, the end-of-block indicator ends the last logical line of input and indicates to the TIP that any queued output can be delivered to your device.

You can control both the end-of-block indicator used and the response used by the TIP; whether the response is sent depends on the value used for the CP command, described earlier in this section. You control the indicator and response through the end-of-block (EB) terminal definition command. This command has the format shown in figure 5-25.

The EB command is valid from console devices in terminal classes 1 through 3, 5 through 8, 10 through 13, and 15 (asynchronous, X.25, and mode 4 terminals). The EB command can also be sent downline for devices in these terminal classes by an application program.

The cpr parameter CR, CL, and LF options cannot be used for mode 4 devices (terminal classes 10 through 13 and 15); the char value for mode 4 devices is the ASCII ETX character and cannot be changed. The end-of-packet sequence (PAD forwarding signal) always acts as an end-of-block indicator for devices connected through an X.25 packet-switching network; if the char parameter is used for an X.25 device, the character chosen is discarded when it is the last character transmitted in a packet sequence.

As indicated in tables 5-1 and 5-2, the default end-of-block indicator and cursor positioning response vary acccording to the terminal class. The range of values permitted for each parameter is given in tables 5-3 and 5-4.

You would normally change the end-of-block indicator used when the current value needs to be stored as data, when you want the current end-of-line indicator to be the end-of-block indicator, or when you want to stop that use of the end-of-line indicator (see the EL command in the previous subsection). You would change the cursor positioning response when the device itself provides part or all of the necessary repositioning, or when receiving a response would be inconvenient or confusing.

For example, suppose you are using the tty asynchronous teletypewriter emulation option of a Wang Model 30 Word Processing System, defined to the network as terminal class 2. The network control character defined for your device is % and the device has been defined as operating in block mode. When transferring a document from the Wang system disk to a CDC CYBER 170, the Wang transmits a carriage return at the end of each text line and an ASCII formfeed code at the end of each text page; however, a page need not end with a carriage return. The CYBER receives each text line as a separate message when that line ends with a carriage return, but not when the line ends with a formfeed code; enroute, the TIP discards the carriage return codes.

You could change the end-of-block indicator to the formfeed code; that action would make the end of a page appear as if the line ended with a carriage return code. You could also define the cursor response for the end-of-block indicator to be "none"; that action would avoid a cursor response at the end of each page. To do this, you could enter the following command before the document transfer begins:

```
%EB= ⊥,NO  [cr]
```

The symbol ⊥ represents the Wang page break code. The page break cannot be entered using the PAGE key during teletypewriter emulation, so this command could have been previously stored in another document and transmitted from that document. (L) could also be used.

ct EB= { eb / eb, cpr / cpr } [cr]

ct Represents the network control character for your console as defined by the CT terminal definition command.

eb Specifies the new end-of-block character; eb can be:

eb	Description
char	Defines a new character for the end-of-block indicator function. This character must differ from the characters currently defined for the AB, BS, B1, B2, CN, and CT command functions.
EB	Selects the default end-of-block character or character sequence for the terminal class.
EL	Selects the character currently defined as the end-of-line indicator.

cpr Specifies the cursor-positioning response to the end-of-block indicator; cpr can be:

cpr	Description
CR	Send a carriage return to the console.
LF	Send a line feed to the console.
CL	Send a carriage return and a line feed to the console.
NO	Send no response; this has an effect similar to declaring a value of N for the CP command, except that the response to an end-of-line indicator is unaffected by this choice.

The parameters eb and cpr are optional and order-independent. If you omit one, the network retains the previous value for the parameter.

[cr] Represents a message termination character, code sequence, or event. Either the end-of-line or old end-of-block indicator can be used.

Figure 5-25. End-of-Block Delimiter Control (EB) Command

You would enter the following command to resume normal operation after the text document is transferred:

```
%EB=EB,YES ⊥
```

or

```
%EB=EB,YES [RETURN] ⊥
```

Again, you could send this command from a previously stored document because you cannot enter the page break using the PAGE key during teletypewriter emulation. When the TIP receives this command, it changes your terminal's end-of-block indicator back to the default value (indicated by the EB parameter) for terminal class 2, and resumes cursor position responses. You must use the default end-of-block indicator, the ASCII end-of-file transmission (EOT) code, after subsequent messages.

Changing Single-Message Mode Transparent Mode Delimiters (DL)

This subsection describes the methods for delimiting single-message input. The next subsection describes the methods for delimiting multiple-message input.

Three events can be independently defined as transparent mode input message and operation delimiters, as follows:

Input of a specific character code from the terminal code set.

An interruption or pause in the input stream from the console to the TIP; such an interruption is called a timeout and must be on the order of 200 to 400 milliseconds.

Transmission of a specific number of terminal code set characters from the console.

Each of the three delimiters is optional and each serves as an end-of-message indicator. At least one delimiter must be in effect before transparent mode input begins. A delimiting character can be the hexadecimal representation of any 6-, 7-, or 8-bit terminal code set character, subject to a restriction described later in this subsection. The delimiting character count can be set to any value from 1 through 4095.

Data received by the Terminal Interface Program after a transparent mode operation delimiter in the same transmission is processed as normalized mode data before transmission to the host. If the delimiter is a character code, the character is always discarded from the transmission to the host and therefore does not occur in either the transparent mode or normalized mode portions of the data.

You use the delimiter (DL) terminal definition command to change the single-message transparent mode delimiters in use. This command has the format shown in figure 5-26.

```
                 Xxx
                 Xxx,Ccount
                 Xxx,TO
ct DL=           Xxx,Ccount,TO    cr
                 Ccount,TO
                 Ccount
                 TO
```

ct — Represents the network control character for your console as defined by the CT terminal definition command.

xx — Specifies the two-digit hexadecimal code of the terminal character you want as an end-of-message indicator and single-message transparent mode operation delimiter. The network does not send the character code as part of the input data.

count — Specifies as an end-of-message indicator and a transparent operation delimiter the decimal value for the number of character codes permitted during single-message transparent mode input; count can be:

$$1 \leq \text{count} \leq 4095$$

TO — Specifies that a timeout of between 200 and 400 milliseconds is both the end-of-message indicator and single-message transparent mode operation delimiter.

cr — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands, earlier in this section.

Figure 5-26. Single-Message Transparent Mode Delimiters (DL) Command

As shown in tables 5-1 and 5-2, the character associated with the terminal's default end-of-line key is a default delimiter for all terminal classes for which transparent mode input is permitted.

Do not use timeout for a delimiter if terminal input is in character mode. (See Data Input Transmissions in section 2.) Timeout would occur after each character of input from a character mode console keyboard, so the timeout delimiter is not meaningful for transmissions longer than one character. You can use a timeout as a delimiter for transmissions from line mode or block mode terminals, where it can occur after the transmission of each line or block to the Terminal Interface Program.

You can successfully use character codes or a character count at character mode, line mode, or block mode terminals. At line mode or block mode terminals, however, occurrence of such a delimiter in the middle of a transmission might cause loss of the first few subsequent characters transmitted as normalized mode data. You can add null characters after the delimiter to avoid this loss.

Note that entering this command with any number of new delimiters cancels all single-message and multiple-message transparent mode input delimiters already in effect. Subsequent transparent mode input is defined to be single-message only. If you specify only one delimiter in the command, you leave the terminal with only one defined delimiter, regardless of the number of delimiters defined before you used the command.

If the only delimiter used is a character code between hexadecimal 80 and FF (inclusive), the parity setting of none or ignore (PA command value of N or I) must be in use before transparent mode input begins. Refer to the PA command later in this section for a description of the difference between the N and I parity options. If the PA value of N or I is not used, the eighth bit in the input character byte is translated as zero by the TIP, and a transparent mode delimiter character code above hexadecimal 7F cannot be detected. Terminals incapable of operating with a parity setting of none must either have a timeout delimiter or character count delimiter declared, or use a character code between 0 and hexadecimal 7F as a delimiter.

Terminal classes 10 through 13 and 15 are configured with the SEND or ETX key (ASCII ETX character or BCD equivalent sequence) as their transparent mode delimiter. Terminal class 4 is configured with the RETURN key (carriage return character) as its transparent mode delimiter. Those configurations cannot change.

Here is an example of transparent mode delimiter use. An electronic typesetting system (ETS) stores document text on magnetic tape cassettes, complete with editing and formatting codes. A publications department using an ETS wants to store several finished documents in a form more compact than multiple tape cassettes provide (on a single nine-track magnetic tape). The ETS is equipped with a communications option that permits it to be operated as a teletypewriter, so the CDC Interactive Facility (IAF) application program can be used to transfer the data.

The ETS is connected to the network by a dialup communication line and logged in to IAF as a device in terminal class 1. The terminal operator uses IAF to create a mass storage file that is later transferred as binary data to magnetic tape. The information placed in the file is transferred from the ETS cassette reader in transparent mode because the cassette contents include carriage return, line feed, and other editing and formatting codes that must be preserved as data for retransmission back to the ETS when the nine-track tape is read.

The terminal operator enters all of the IAF commands needed to set up the file for input, then sets up the TIP for transparent mode input in the following manner:

```
%DL=TO  cr
```

The network control character % used in this entry is defined by the site for the console being used. This command establishes one and only one single-message transparent mode delimiter for the console. Timeout is chosen, because the two default delimiters for terminal class 1 (a character count of 2043 and the carriage return character code) will not produce the results needed.

Transmissions from the ETS tape unit are continuous and are not divided into units of 2043 characters; to avoid ending transparent mode and losing characters in the middle of a unit of text transmission, the character count delimiter cannot be used. Part of the data that must be preserved by transparent mode use is the embedded carriage return character codes; so that these codes will not be lost as discarded delimiter characters, the default character code delimiter cannot be used.

The timeout delimiter is used because the tape unit automatically stops at the end of a text unit. This action produces a pause in input at a convenient point for terminal operator control of the transfer.

Transmission is begun by entering the command:

%IN=X [cr]

which is described later in this section, then pressing the console control keys that start the cassette reader and transmit the cassette's contents. Each time the tape unit stops, the Terminal Interface Program returns automatically to normalized mode. The terminal operator repeats this procedure until the cassette contents are completely transferred.

Changing Multiple-Message Mode Transparent Mode Delimiters (XL)

This subsection describes the methods for delimiting multiple-message input. The previous subsection describes the methods for delimiting single-message input.

You use the extended-delimiter (XL) terminal definition command to establish the multiple-message transparent mode and message delimiters used when servicing the console. This command has the format shown in figure 5-27.

As shown in tables 5-1 and 5-2, no default exists for any terminal class from which transparent mode input is permitted. No default delimiters exist because all terminal classes are initially defined for single-message input.

Two events can be independently defined as transparent mode input operation delimiters, as follows:

- Input of two specific character codes (or one code twice) from the terminal code set.
- An interruption or pause in the input stream from the console to the TIP; such an interruption is called a timeout and must be on the order of 200 to 400 milliseconds.

Both mode delimiters are optional, but at least one must be in effect before transparent mode input begins. Either mode delimiter also indicates the end of the last transparent input message.

```
ct XL= { Xxx
       { Xxx,Ccount
       { Xxx,Yyy
       { Xxx,Ccount,Yyy
       { Xxx,Ccount,TO,Yyy
       { Xxx,TO,Yyy
       { Xxx,Ccount,TO
       { Xxx,TO
       { Ccount,TO
```

[cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command.

xx — Specifies the two-digit hexadecimal code of the terminal character you want as an end-of-message indicator. The network does not send the character code as part of the input data. If you do not specify a value yy, the occurrence of two successive xx characters acts as a delimiter for transparent input mode operation; otherwise, the network allows more transparent mode input after this message is sent to the host application program.

count — Specifies as an end-of-message indicator the decimal value for the number of character codes permitted in one message; count can be:

$1 \leq count \leq 4095$

The network allows more transparent mode input after this message is sent to the host application program.

TO — Specifies that a timeout of between 200 and 400 milliseconds is the end-of-message indicator for the last message and delimits multiple-message transparent mode operation.

yy — Specifies the two-digit hexadecimal code of the terminal character you want as an end-of-message indicator and transparent mode input operation delimiter. The character with the code yy is a delimiter for transparent input mode when it follows the character represented by the code xx. The network does not send either character code as part of the input data. After receiving the code sequence xx and yy, the network treats subsequent input as normalized mode data. If you do not specify a value yy and specify xx, the occurrence of two successive xx characters ends transparent input mode.

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands, earlier in this section.

Figure 5-27. Multiple-Message Transparent Mode Delimiters (XL) Command

Data received by the Terminal Interface Program after a transparent mode operation delimiter is encountered in the same transmission is processed as normalized mode data before being sent to the host. If the delimiter is a character code, the character is always discarded from the transmission to the host and therefore does not occur in either the transparent mode or normalized mode portions of the data.

Two events can be independently defined as transparent mode end-of-message indicators (message-forwarding signals), as follows:

- Input of a specific character code from the terminal code set
- Transmission of a specific number of terminal code set characters from the console

When an end-of-message indicator is encountered, the TIP forwards the message to the host and waits for further input.

A character used as a delimiter or end-of-message indicator can be the hexadecimal representation of any 6-, 7-, or 8-bit terminal code set character, subject to a restriction described later in this subsection. If you use the Y option, it must follow the X option (otherwise, the Y option is ignored). You must specify at least one of X, Y, or TO to select a transparent input mode delimiter.

The delimiting character count can be set to any value from 1 through 4095.

You can use timeout as a delimiter for transmissions from block mode terminals, where it can occur after the transmission of each block to the Terminal Interface Program. You can also use timeout for transmissions from line mode terminals, when you do not know the length of a line in advance and all character codes must be preserved as part of the input data.

You can successfully use character codes or a character count as delimiters or an end-of-message indicator at character mode, line mode, or block mode terminals. If you define the end-of-message indicator (X option character) to be the end-of-line indicator defined by the EL command and the input mode delimiter character (Y option character) to be the end-of-block character defined by the EB command, transparent mode input operation will be very similar to full-ASCII normalized mode.

Entering the XL command with any number of new parameters cancels all single-message and multiple-message transparent mode input delimiters and end-of-message indicators already in effect. Subsequent transparent mode input is defined to be multiple-message only. For example, if you specify only one parameter in the command, you leave the terminal with only one defined delimiter, regardless of the number of delimiters or end-of-message indicators defined before you used the command.

If the only delimiter used is a character code between hexadecimal 80 and FF (inclusive), the parity setting of none or ignore (PA value of N or I) must be in use before transparent mode input begins. Refer to the PA command later in this section for a description of the difference between the N and I parity options. If the PA value of N or I is not used, the eighth bit in the input character byte is translated as zero by the TIP and a transparent mode delimiter character code above hexadecimal 7F cannot be detected. Terminals incapable of operating with a parity setting of nonemust have a timeout delimiter declared, or use a character code between 0 and hexadecimal 7F as a delimiter.

Terminal classes 10 through 13 and 15 are configured with the SEND or ETX key (ASCII ETX code or BCD equivalent) as their transparent mode delimiter. Terminal class 4 is configured with the RETURN key (carriage return code) as its transparent mode delimiter. Those configurations cannot change.

Here is an example of transparent mode delimiter and end-of-message indicator use. An electronic typesetting system (ETS) stores document text on magnetic tape cassettes, complete with editing and formatting codes. A publications department using an ETS wants to store several finished documents in a form more compact than multiple tape cassettes provide (on a single nine-track magnetic tape). The ETS is equipped with a communications option that permits it to be operated as a teletypewriter, so the CDC Interactive Facility (IAF) application program can be used to transfer the data.

The ETS is connected to the network by a dialup communication line and logged in to IAF as a device in terminal class 1. The terminal operator uses IAF to create a mass storage file that is later transferred as binary data to magnetic tape. The information placed in the file is transferred from the ETS cassette reader in transparent mode because the cassette contents include carriage return, line feed, and other editing and formatting codes that must be preserved as data for eventual retransmission back to the ETS when the nine-track tape is read.

The terminal operator enters all of the IAF commands needed to set up the file for input, then sets up the TIP for transparent mode input in the following manner:

```
%XL=X17,Y04  cr
```

The network control character % used in this entry is defined by the site for the console being used. This command establishes one and only one end-of-message indicator multiple-message transparent mode delimiter for the console. The code sequence of an ASCII end-of-transmission block (ETB) followed by an ASCII end-of-transmission (EOT) is chosen. (A character count will not produce the results needed.)

Transmissions from the ETS tape unit are essentially continuous but are not divided into units of a fixed number of characters; to avoid leaving transparent mode and losing characters in the middle of a unit of text transmission, the character count delimiter must be avoided. The ETS sends the ETB code and stops at the end of each text unit on the tape. When the tape ends, the ETS sends an ETB code and the EOT code. To send all of the text units on a cassette tape, the terminal operator restarts the tape reader manually each time it stops.

Transmission is begun by entering the command:

```
%IN=X  cr
```

which is described later in this section, then pressing the console control keys that start the cassette reader and transmit the cassette's contents. Each time the tape unit stops transmitting, the Terminal Interface Program remains in transparent mode and the terminal operator repeats the procedure until the cassette contents are completely transferred.

CONTROLLING PARITY AND CHARACTER ECHOING

Two terminal definition commands allow you to change how the Terminal Interface Program handles the functions of parity processing and character echoing. The PA command allows you to change parity bit processing for your console when more than one possibility exists. The EP command determines whether the TIP echoes (echoplexes) your input characters back to your terminal as you enter them.

Changing Parity Processing (PA)

When a terminal is serviced in either transparent mode or normalized mode, it sends and receives character data bytes six, seven, or eight bits long. Bytes of the first two sizes can also be accompanied by a parity bit. The parity bit is used by the terminal and the Terminal Interface Program in an arithmetic test to ensure that the bits received for the character are the same bits sent.

If the bit exists and is used to make the sum of the other bits odd or even, then the terminal operates with odd or even parity. If the bit exists but is not used for odd or even summation, the terminal operates with zero parity. If the bit does not exist, then all bits sent and received are part of the data character byte and the terminal operates with no parity. The Terminal Interface Program always performs the parity processing defined to it as appropriate for the terminal.

You can change the parity processing performed by the TIP for both input and output character codes through the parity (PA) terminal definition command. The format of this command is shown in figure 5-28.

The PA command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous terminals). This command can also be sent downline for consoles in those terminal classes by a network application program.

As shown in tables 5-1 and 5-2, the default parity setting for terminal classes 1 through 3 and 5 through 8 is even. The default input parity setting for X.25 devices in terminal classes 1 through 3 and 5 through 8 is zero; the TIP does not check input parity from X.25 network devices.

The default and only possible parity setting for devices in terminal classes 4, 10 through 13, and 15 (2741-compatible and mode 4 terminals) is odd. Terminals in terminal classes 9, 14, 16, 17, and 18 (HASP and bisynchronous terminals) use the 8-bit character code EBCDIC, and therefore operate without a separate parity bit; parity processing does not occur for these terminals.

ct PA=pa [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command.

pa — Specifies the type of parity processing needed by your device; pa can be:

pa	Description
E	Set parity bit for even parity during output; check for even parity (except X.25 devices) and set parity bit to zero on input.
I	In transparent mode, do not set or check the parity bit because it is part of the input or output data. Ignore the parity bit when checking for the transparent end-of-message delimiting character. In normalized mode, set the parity bit to zero on input and output.
N	In transparent mode, do not set or check the parity bit because it is part of the input or output data. In normalized mode, set the parity bit to zero on input and output.
O	Set parity bit for odd parity during output; check for odd parity (except X.25 devices) and set parity bit to zero on input.
Z	Set parity bit to zero on input and output.

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands, earlier in this section.

Figure 5-28. Parity Processing Selection (PA) Command

When a terminal is serviced in normalized mode, all character bytes sent or received by the host software comprise seven bits of ASCII character code and a zero parity bit as the eighth bit in the 8-bit character byte. When a terminal is serviced in transparent mode, however, the character code in the 8-bit character byte is the 6- or 7-bit terminal character code, plus a separately processed parity bit.

Because the TIP processes parity separately from character code translation, it is possible for an application program to send or receive transparent mode character codes without the parity bit. To send or receive 8-bit codes, however, requires a parity setting of none or ignore to be in effect.

Table 5-7 summarizes the parity processing performed by the TIP for each of the four parity settings. This table also shows the dependencies on the transmission mode in use. You must remember these dependencies when changing the parity processing defined for the terminal.

Transmissions in normalized mode can cause data to appear garbled at the terminal by declaration of a PA value incompatible with the physical terminal, but upline and downline character code conversions are not otherwise affected by the value declared. Transmissions in transparent mode can be garbled or codes changed by declaration of a PA value incompatible with the physical terminal.

You would normally change the parity processing for the terminal only when:

- The initial value is incorrect for your terminal
- The parity bit portion of the terminal character code must be treated as part of the meaningful data

The latter occurs for transparent mode transmissions only, and usually consists of switching the processing in effect from odd, even, or zero to none or ignore (then back to odd, even, or zero).

For example, if you are using a CDC 110 personal computer with a CDC 721 Viking Extended console (terminal class 3), you might want to create a mass storage file at the host site of canned character output data containing all graphic and control character bytes in the terminal's graphic option code set. You can do this through the CDC Interactive Facility (IAF) using transparent mode transmissions to store data treated by IAF as binary mode data.

Because the eighth bit of the data is used by the Viking console to identify special characters, you must enter the data without parity processing, so that the parity bit of each character byte is preserved when stored. The storage operation could be accomplished from the terminal keyboard by the following set of entries:

```
%PA=I  cr
%XL=X0D,Y04  cr
%IN=X  cr
data  cr
 (D)
%PA=E  cr
```

You can use these entries together with commands to IAF that create and save the file containing the transmitted data. In this series of entries, you change the parity processing to ignore before

TABLE 5-7. PARITY BIT PROCESSING TRANSMISSION MODE DEPENDENCIES

Event	Parity Defined						
	0 Both Modes	E Both Modes	Z Both Modes	N Normalized	N Transparent	I Normalized	I Transparent
Terminal code input parity checked	Yes (except X.25)	Yes (except X.25)	No	No	No	No	No
Terminal code output parity bit set or cleared	Yes	Yes	Yes	No	No	No	No
Character code bit 7 (the eighth bit) set to zero for application program input	Yes	Yes	Yes	Yes	No	Yes	No
Application program output character code bit 7 (the eight bit) set to zero before translation to terminal code	Yes	Yes	Yes	Yes	No	Yes	No
Processing omitted (8-bit data with no parity bit)	No	No	No	No	Yes	No	Yes
Number of significant bits in the end-of-message or transparent mode input delimiting character	7	7	7	na	8	na	7

transparent mode input begins. You define the end-of-message indicator as a carriage return and the end-of-transparent mode delimiter as an EOT code (CONTROL and D). You enter transparent input mode with the IN command to the TIP (this command is described later in this section).

You enter the transparent mode data from the CDC 110's disk drive and end it when the CONTROL and D keys are pressed. The final entry restores the parity processing to normal to ensure accurate transmission of subsequent data.

Changing Echoplex Operation (EP)

Because input and output are logically separate operations, it is possible for a terminal to generate input data without any visual confirmation from the network of the data entered. Line mode and block mode terminals normally operate in this manner; input transmission is only visible as cursor as the screen content is scanned and sent. Such terminals usually perform internal character echoing so that the input data can be edited before transmission. To edit the line, it must be seen. Such terminals do not require character echoing by either an external modem or the Terminal Interface Program.

The method used at character mode consoles to confirm input data is to return each character entered to the terminal output mechanism as the character is entered. This character echoing can be performed by the terminal hardware itself, by an internal or external modem or acoustic coupler connecting the terminal to the communication line, or by the Terminal Interface Program.

Character mode terminals sometimes depend on external character echoing. The modems commonly used with such terminals perform the echoing needed. However, it is possible for these terminals to use modems that do not perform character echoing. To make use of such terminals and modems easier, the Terminal Interface Program can be instructed to perform the character echoing operation.

You use the echoplex (EP) terminal definition command for this purpose. This command has the format shown in figure 5-29.

The EP command is valid from console devices in terminal classes 1 through 3 and 5 through 8 (asynchronous terminals). If the device is connected through an X.25 packet-switching network, the TIP informs the X.25 network packet assembly/disassembly (PAD) service to perform the character echoing operation. This command can also be sent downline to consoles in those terminal classes by an application program.

As indicated in tables 5-1 and 5-2, all terminals of terminal classes 1 through 3 and 5 through 8 (asynchronous terminals) use a default selection of no character echoing by the TIP, because this is the most common possibility. Terminals in terminal classes 9 through 18 (HASP, mode 4, and bisynchronous terminals) use communication protocols that

```
ct EP [ =option ] [cr]

ct      Represents the network control character for your console as defined by the
        CT terminal definition command.

option  Indicates whether input echoing should occur; option can be:

            Y   yes (default if =option is omitted)

            N   no

[cr]    Represents a message termination character, code sequence, or event.
        Either the end-of-line or end-of-block indicator can be used; refer to the EL
        and EB commands, earlier in this section.
```

Figure 5-29. Echoplexing (EP) Command

make character-by-character echoing impossible, so character echoing cannot be selected for such terminals.

You should be careful when changing echoplex operation. The following paragraphs explain the effects of the option chosen.

If the terminal or modem performs its own character echoplexing:

- Each entry of a character at the console causes the character to appear twice (or be overwritten) on the output mechanism when an EP value of Y is used.
- Each entry of a character causes the character to appear once on the output mechanism when an EP value of N is used.

If the terminal or modem does not perform echoplexing:

- Each entry of a character causes the character to appear once on the output mechanism when an EP value of Y is used.
- No indication of a character entry appears on an interactive output device when an EP value of N is used.

Modems with a duplex switch normally perform character echoing when the switch is in the HALF position but do not perform it when the switch is in the FULL position. Full-duplex modems that cannot be switched to half duplex operation, however, can be used for half duplex operation. If your terminal does not perform character echoing for itself, it can use such a modem and a half-duplex communication line mode after you enter the following terminal definition command:

```
%EP=Y [cr]
```

The site defines the network control character as % for your console. Before you press the transmission key for this entry, nothing you enter at the console appears on the screen. After the TIP acknowledges the entry with a line feed code, all your subsequent entries are returned to the console by the TIP.

You should distinguish between setting the duplex switch for an internal terminal modem and setting the duplex switch on an external acoustic coupler. Acoustic couplers can also perform character echoing based on a duplex switch setting. When a full-duplex external acoustic coupler is used and character echoing occurs but is not performed by the TIP, the terminal possesses an internal modem. Internal modems usually have separate duplex switches on the terminal housing. In this case, all switch settings should be compatible with the EP value in use.

It is preferable to have your terminal or modem do the character echoing, rather than have the network software do it. When the network software does the character echoing, there can be a perceptible pause at lower speed terminals between pressing a key and seeing the character appear on the screen. This pause can be distracting to an experienced typist.

IDENTIFYING CONSOLE MECHANISMS

Because input and output are separate operations, the Terminal Interface Program allows you to separately specify the console mechanism used for each operation. The input (IN) command permits you to identify your input mechanism so that the TIP can properly service that mechanism, while the output (OP) command provides the same function for your output mechanism.

Selecting an Input Mechanism and Transmission Mode (IN)

You can decide to perform input through a paper tape reader (or an emulation device for a paper tape reader), or through a keyboard or other mechanism. You can also decide to perform input in transparent mode or in normalized mode. In normalized mode, you can choose between block mode for entries and the network's line mode for entries.

Input from paper tape mechanisms require the TIP to process character codes differently than when input from other devices. Input in normalized mode requires the TIP to perform the translation and editing functions described earlier in this manual. You therefore must tell the TIP which of the following combinations of these input options is applicable:

Input is from paper tape and is in normalized block mode.

Input is from paper tape and is in transparent mode.

Input is not from paper tape and is in normalized block mode.

Input is not from paper tape and is in normalized line mode.

Input is not from paper tape and is in transparent mode.

By convention, input that is not from a paper tape reader is associated with a keyboard or a mechanism that simulates a keyboard. Depending on the way a site configures the console, the terminal might begin its access to the network with a paper tape input mechanism assumed; this is not the most common situation. A terminal must always begin access to the network in normalized mode.

For either of the transparent modes, you must establish the transparent message and operation delimiter (DL or XL) before you enter transparent mode, or the single-message delimiters established by the network administrator will be used. Keyboard input is allowed in either of the paper tape modes.

The detailed effects of the input option currently in use are described in the following paragraphs. Whenever possible, the TIP detects the end of transparent input and automatically changes option combinations for you. This automatic redefinition switches the terminal back to keyboard normalized mode from keyboard transparent mode or to paper tape normalized mode from paper tape transparent mode. The other switches cannot be made automatically, and a terminal definition command is provided so that you can make them manually.

You use the input device (IN) terminal definition command to change the combination of input options used for TIP processing of transmissions. This command has the format shown in figure 5-30.

As indicated in tables 5-1 and 5-2, the default combination of options for consoles in all terminal classes is normalized mode input from a keyboard. The IN command is valid from console devices in terminal classes 1 through 8 (asynchronous terminals), 10 through 13, 15 (mode 4 terminals), and 18 (3270 terminals).

Devices in terminal class 4 (2741-compatible terminals) can use only the KB, X, and XK options; mode 4 terminal consoles (classes 1 through 13 and 15) are always in block mode and can use only the X option. The IN command can also be sent downline for consoles in these terminal classes by an application program.

When the IN value XK, XP, or X is used, data entered subsequently from the console is sent to the host as transparent mode messages. Transparent mode operation continues for input until a transparent mode operation delimiter previously defined for the device is detected; detection of the delimiter causes the TIP to return automatically to normalized mode operation.

When an IN value of XK is used, the TIP returns automatically to normalized mode operation with an IN value of KB. When an IN value of XP is used, the TIP returns automatically to normalized mode operation with an IN value of PT. When an IN value of X is used, the TIP returns automatically to whichever normalized mode alternative was previously in effect.

When you use the IN value of PT, the Terminal Interface Program treats the paper tape reader as a block mode device with a fixed end-of-block indicator character (the ASCII DC3 code, representing the X-OFF character). Whenever the TIP expects input from the device, it sends and ASCII DC1 code as an X-ON character to start the paper tape reader. The code is issued after the end of a message is

ct IN=in [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

in — Specifies the input mechanism and transmission mode of your device; in can be:

in	Description
BK	Selects block mode (as from a form terminal) and normalized input mode.
KB	Selects the keyboard in line mode and normalized input mode.
PT	Selects the paper tape reader and normalized input mode (not valid for IBM-2741-compatible devices).
X	Selects the current input device and transparent input mode.
XK	Selects the keyboard and transparent input mode.
XP	Selects the paper tape reader and transparent input mode (not valid for IBM- 2741-compatible devices).

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands, earlier in this section.

Figure 5-30. Input Device Description (IN) Command

output to the terminal and the message empties the terminal's output queue.

X-OFF code (ASCII DC3) processing by the Terminal Interface Program is the primary difference in input data treatment between IN value declarations of a keyboard input source or declarations of a paper tape reader source. When the IN value PT or XP (or X after PT) is used, an X-OFF code embedded in the input is sent to the host as part of the data. However, if the X-OFF code follows an end-of-line or end-of-block indicator character, the TIP does not send the X-OFF code upline as data. An X-OFF code entered after using one of the keyboard input declarations is always sent upline as part of the data.

Because transparent mode input is neither translated nor inspected by the TIP prior to transmission to the host, you cannot switch the terminal from keyboard transparent mode to paper tape transparent mode directly. Also, you cannot switch from paper tape transparent mode directly to keyboard transparent mode. In both cases, the terminal definition command to perform the switch would be contained within the transparent mode transmission and would therefore be sent to the host as data instead of being edited out of the input.

You can change from paper tape normalized mode to keyboard normalized mode during input transmissions by punching the appropriate terminal definition command into the paper tape at a point where tape reading will resume following an X-ON code from the TIP. You can also make this change by entering the terminal definition command from the keyboard after the reader has finished reading the current paper tape.

Examples of paper tape input are given in section 3 for normalized mode data. Support of transparent mode paper tape input depends on the application program; no meaningful examples of transparent mode input using paper tape can be given for standard network application program communication.

Selecting an Output Mechanism (OP)

If you decide to record data through a paper tape punch or paper tape emulator device, the TIP provides additional formatting characters for the output. If you do not choose to record output data on paper tape, no additional formatting characters are supplied.

You cannot make the choice of transparent mode or normalized mode for output data directly, as explained in section 2. This choice is made by the network application program, but not necessarily in response to a command from you.

Output to paper tape requires the TIP to provide character codes not required in output to other devices. You should therefore inform the TIP when paper tape is being used for output.

You inform the TIP via an output mechanism (OP) terminal definition command. The format of the OP command is shown in figure 5-31.

This command is valid from console devices in terminal classes 1 through 8 (asynchronous and X.25 terminals); the PT option is not allowed from devices in terminal class 4 or devices connected through an X.25 packet-switching network. The command can also be sent downline for consoles in these terminal classes by an application program.

As indicated in tables 5-1 and 5-2, the default output mechanism for consoles in all terminal classes is a device other than a paper tape punch.

The value you declare for OP determines whether the Terminal Interface Program supplies carriage return and line feed codes at page boundaries (see the PW command). The value declared for OP also determines whether the TIP automatically supplies an X-ON or an X-OFF code after the end of a downline normalized mode message; the latter occurs only when a paper tape punch is declared as the output mechanism and a post-print format effector has been included for a logical line.

ct OP=op [cr]

ct — Represents the network control character for your console as defined by the CT terminal definition command. You must press the ATTN key on a 2741-compatible device before entering this character.

op — Specifies your device's output mechanism; op can be:

op	Description
DI	Specifies a display.
PR	Specifies a printer.
PT	Specifies a paper tape punch (not valid for X.25 or IBM-2741-compatible devices).

[cr] — Represents a message termination character, code sequence, or event. Either the end-of-line or end-of-block indicator can be used; refer to the EL and EB commands, earlier in this section.

Figure 5-31. Output Device Description (OP) Command

Output can be produced on a paper tape after you declare the OP value DI or PR, but the application program must embed any desired X-OFF codes in the last block of output.

The OP command gives you a method of switching to and from paper tape punch use. It also provides a method of switching from printer-formatted output to display-formatted output or vice versa. You can use this command to change page boundary processing by the TIP.

When a value of DI is used, the TIP calculates page boundaries (see the PG and PL commands) but does not perform output formatting when the page width is reached. The TIP does insert idle characters as necessary for the device's definition when the page width is reached.

When a value of PR is used, the TIP calculates page boundaries and inserts carriage returns, line feeds, and idle characters as necessary for the device's definition when the page width is reached. (Refer to the PW command, earlier in this section.)

Examples of paper tape output are given in section 3 for normalized mode data. Support of transparent mode paper tape output depends on the application program; no meaningful examples of transparent mode output using paper tape can be given for standard network application program communication.

THE COMMUNICATIONS CONTROL PROGRAM AND THE NETWORK PROCESSING UNIT

This section contains a general description of how the Communications Control Program (CCP) functions in a network processing unit (NPU). This description:

- Introduces the major hardware components of the NPU that form the hardware environment for CCP operation.
- Briefly describes the multiplex subsystem and the mechanics of transferring data in the two general formats used by the network; these are the interactive virtual terminal (IVT) format defined in section 2, and the physical record unit (PRU) format used to support CDC-defined batch devices through the special batch data interface mentioned in section 2.
- Summarizes the operation of CDC-written Terminal Interface Programs (TIPs).

You do not need to read this section to use network features. It is provided as background information, and might be useful when trying to use more sophisticated features of a terminal or of a personal computer. It can help you determine how to set up a terminal or personal computer for correct communication through a CDC network.

HARDWARE ENVIRONMENT

Each CDC network processing unit consists of:

- A 2551 series Communications Processor, with macromemory and micromemory
- A multiplex subsystem, containing varying numbers and types of communications line adapters
- Zero, one, or two CYBER channel couplers

Figure 6-1 shows the maximum architecture of such a network processing unit. Some NPUs also have a magnetic tape cassette drive and an operator's console that is not part of the network.

2551 SERIES COMMUNICATIONS PROCESSOR

The 2551 series Communications Processor has both a programmable micromemory and a macromemory; each model of the 2551 series has varying amounts of macromemory, with two sizes of random access micromemory (RAM) logic. 8192 words (8K) of micromemory are required to run the current release of CCP.

Because models can be modified on site, a model designation does not necessarily correspond to a specific macromemory or micromemory configuration. The internal characteristics of a specific 2551 Communications Processor model are described in the corresponding hardware reference manual listed in the preface.

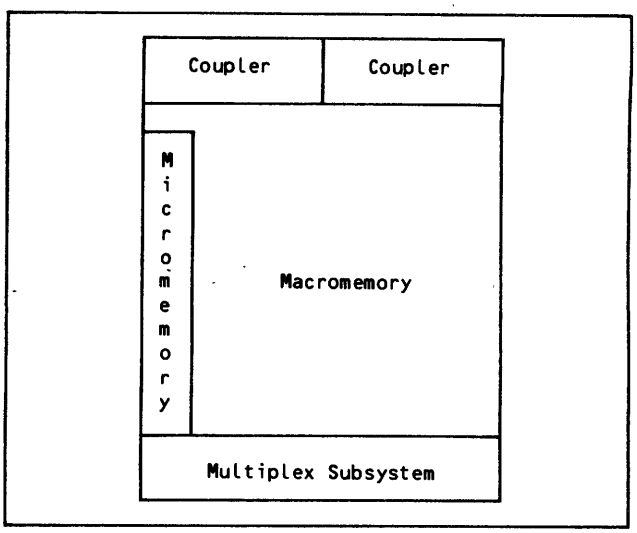


Figure 6-1. Basic Components of a CDC Network Processing Unit

Each NPU might require its own variant of CCP, depending on the macromemory size of the unit and the software modules that need to reside in the unit. Variants are generated as described in the NOS Version 2 Analysis Handbook described in the preface. These variants reside in the host NPU load file (NLF). The appropriate variant for loading into each NPU is determined by the host Network Supervisor program from information contained in the network configuration file. This process is described in the Network Access Method Version 1 Network Definition Language Reference Manual listed in the preface.

CYBER CHANNEL COUPLER

Local NPUs (front-end NPUs for CDC host computers) have at least one CYBER channel coupler. Although up to two channel couplers can be physically housed in an NPU cabinet, the channel coupler is treated as an external device by the NPU. The coupler contains programmable registers; it serves as a transfer mechanism to convert the 8-bit data bytes within one 16-bit NPU memory word into 8-bit data bytes within two 12-bit words of a CDC CYBER 170 computer's peripheral processor.

CCP services the coupler through a software module called the Host Interface Program (HIP). The external characteristics of the CYBER channel coupler are described in the Network Access Method Version 1 Network Definition Language Reference Manual mentioned in the preface. The internal characteristics of a coupler are described in the appropriate 2551 Communications Processor model hardware reference manual listed in the preface.

CASSETTE DRIVE

The magnetic tape cassette drive and accompanying deadman timer hardware (collectively called the system autostart module) are required if the 2551 has two couplers or is used as a remote NPU. When an NPU has a cassette drive, the drive is used to begin the loading of CCP from a host computer.

The cassette drive loads a copy of the system autostart module program (SAM-P) from a CDC-supplied cassette. SAM-P is essentially a bootstrap loader that obtains the copy of CCP appropriate for its NPU from a copy of the Network Supervisor program in a host computer.

The cassette drive can also be used to load optional off-line hardware diagnostic software for use with the offnet NPU console.

NPU CONSOLE

If the NPU has an operator's console, that console is not part of the network; it connects to a special port of the NPU and is not serviced through a communications line adapter. This offnet console is not the console associated with the NPU operator who is known to the Communications Supervisor program in the host as a network operator or NOP.

The offnet NPU console is used to run optional on-line or off-line diagnostic software; use of that software is described in the Communications Control Program Version 3 Diagnostic Handbook listed in the preface. If the site modifies its copy of CCP, the offnet console can be used with the internal Test Utility Package for on-line debugging of the code.

MULTIPLEX SUBSYSTEM

The multiplex subsystem contains the following components:

- A multiplex loop interface adapter (MLIA)
- One or more loop multiplexers (LMs)
- One or more communications line adapters (CLAs)
- Related firmware (microprograms) and software

The basic relationships of these components are shown in figure 6-2.

There are three types of CDC communications line adapters:

- Model 2560 series synchronous CLAs, used to support lines connecting Mode 4 protocol, IBM 2780 or IBM 3780 bisynchronous protocol, the IBM 3270 bisynchronous protocol, or HASP protocol terminals to the NPU

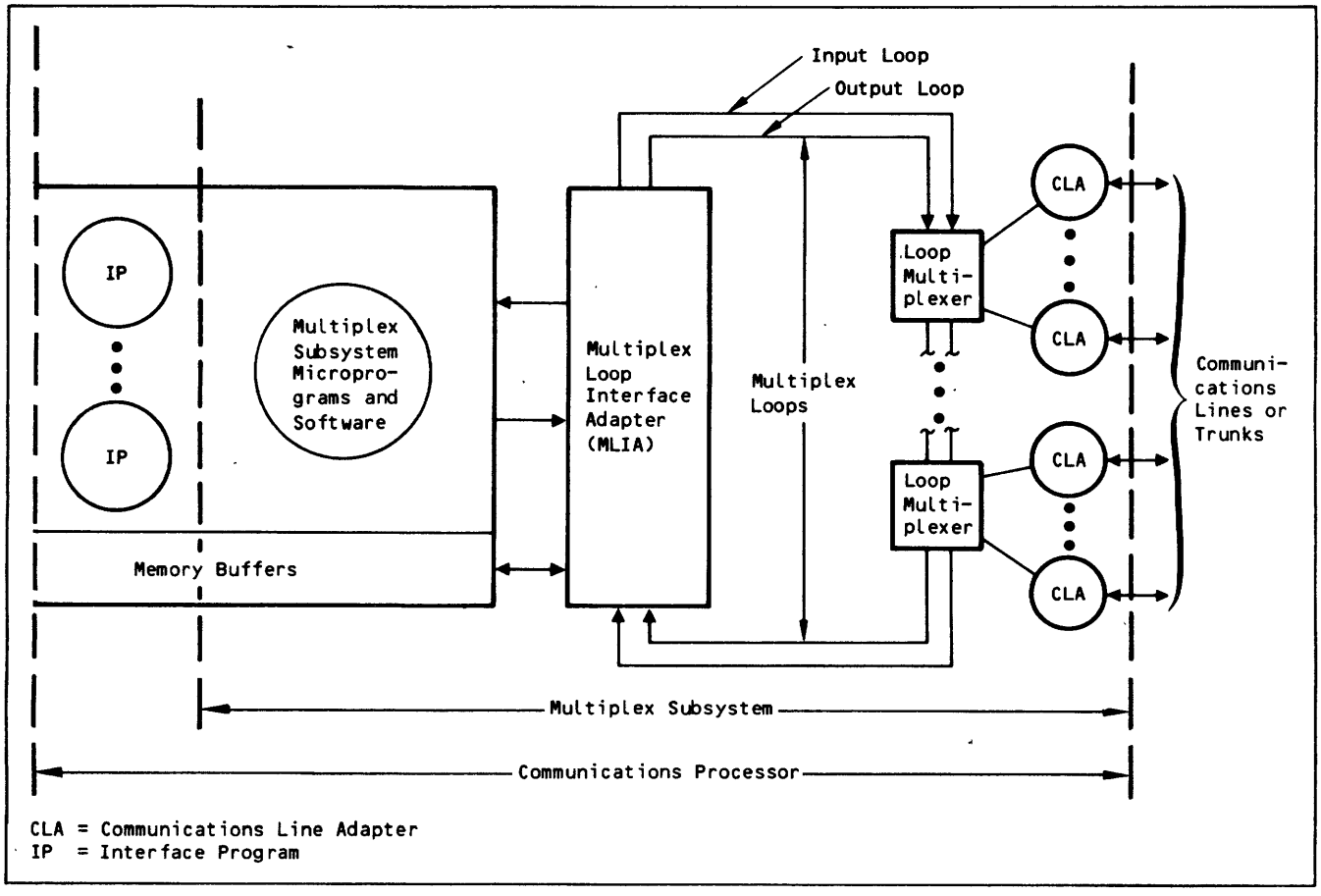


Figure 6-2. Basic Components of the Multiplex Subsystem

Model 2561 series asynchronous CLAs, used to support lines connecting teletypewriter-compatible or IBM 2741-compatible terminals to the NPU

Model 2563 series high-level (synchronous) data link control (HDLC or SDLC) CLAs, used to support trunks connecting other NPUs or lines connecting asynchronous terminals or other hosts to the NPU through X.25 packet-switching networks or a direct link to another NPU

The types of CLAs in the multiplex subsystem of a specific NPU depend on the types of connections that NPU must support. The external characteristics of CLAs are described in the Network Access Method Version 1 Network Definition Language Reference Manual mentioned in the preface. The internal characteristics and functions of CDC-supplied CLAs are described in their respective hardware reference manuals listed in the preface.

MULTIPLEX SUBSYSTEM OPERATION

The following paragraphs provide a brief overview of the flow of data through all of the hardware and software components listed above.

The multiplex subsystem has two principal tasks:

Relieves the TIPs of having to process communication line data according to physical line characteristics (most line characteristics are invisible to the TIPs as a result of multiplex subsystem processing)

Multiplexes data to match the low-speed characteristics of individual terminals with the high-speed characteristics of the NPU and the host

The multiplex subsystem:

Receives serial data from communications lines, places it in a circular input buffer, demultiplexes it, and places it into a separate input buffer for each device on a communication line

Detects and processes special characters

Translates code on input

Detects breaks in communication

Checks the cyclic redundancy count (CRC)

Checks and generates character parity

Controls modems on the NPU end of communication lines and analyzes line status

Processes commands issued by the TIPs

Chains input data buffers as necessary to assemble network data blocks

Transmits serial data to communications lines from communications line output buffers

INPUT MULTIPLEXING

Each line has a communications line adapter (CLA). If a character is ready when each CLA is sampled, the CLA places the character on the input multiplex loop together with information identifying the source (line) and in some cases, the nature of the character (for instance, a character byte can contain modem signal information rather than a character of data). All information on the input multiplex loop is routed to a circular input buffer (CIB), which is usually 512 words long. The demultiplexing operation picks data from this buffer and reconstitutes the data input lines separately for each communication line.

INPUT DEMULTIPLEXING

All TIPs except the HASP and BSC TIPs and possibly a site-written TIP process data in a single pass. For one-pass TIPs, the multiplex subsystem is responsible for selecting data from the CIB as well as for putting it into a communication line input buffer.

When transmission block input starts, the multiplex subsystem reserves a data buffer for the network block (data buffers are normally 64 words long, with two characters per NPU word). The words in the CIB are identified by the communication line number and are packed into an input buffer for that line. If a buffer is filled before the transmission block is complete, another buffer is assigned and is chained to the first.

When the state programs detect the end of input, the TIP appropriate for the terminal type is notified. The TIP continues the processing by passing control of the data to the BIP, which formats the data into network blocks and controls the routing of those blocks. The BIP passes control of the network blocks to the Host Interface Package (HIP) or the Link Interface Package (LIP).

The LIP passes the network block across a trunk communications line to another LIP, as described below under Trunk Multiplexing. The HIP passes the network block through the coupler to the PPU of the host.

OUTPUT MULTIPLEXING

When the NPU has received a network data block from the host and a TIP has transformed the code and/or format from IVT or PRU to terminal format, the TIP notifies the multiplex subsystem that the transmission block is ready for output. The multiplex subsystem picks the characters from the communication line's output buffer one at a time, whenever a new output data demand (ODD) is generated by the CLA. (The ODD indicates that the communications line is ready to receive another character.)

The outgoing characters are placed on the output multiplex loop, along with such control characters as are needed, and an address that will be recognized by the CLA for the active line to that terminal device. The CLA recognizes the address, picks that data from the output loop, and marks it as

being sent. When the contents of the entire output buffer for the communications line have been transmitted, and the transmission block has been received by the terminal, the multiplex subsystem can notify the TIP (or the TIP can wait for an acknowledgment response from the terminal). The TIP releases the output buffer and notifies the host of a successful transmission.

OUTPUT DEMULTIPLEXING

The demultiplexing of downline transfers is a terminal function; that is, the network block is reconstituted in a buffer for the terminal. That buffer becomes a transmission block to the terminal, which must then subdivide or route the block for delivery to the proper output device.

TRUNK MULTIPLEXING

If a remote NPU is included in the network, transmissions between the local NPU and the remote NPU take place over a trunk communications line. In the local NPU, a link interface package (LIP) sets up the output buffer, and the network block in that buffer remains in IVT or PRU format while being transmitted over the trunk. In the remote NPU, the downline network block goes through the CIB and is then demultiplexed by the LIP.

The LIP passes the block to the BIP, which performs the operations described under Input Multiplexing. The BIP then passes the block to the appropriate TIP.

After the TIP in the remote NPU translates the code from IVT or PRU format to terminal format, the network block is treated as an output network block in a local NPU; that is, the block is multiplexed for transmission, is sent, and is acknowledged. The acknowledgment must be formatted as any other input network block: remultiplexed and sent upline through the local NPU to the host.

Input network blocks (upline traffic) are reformatted from terminal to PRU or IVT format as in a local NPU, but are then sent by the LIP through the remote NPU multiplexer, received by the local NPU multiplexer, and reconstructed into complete network blocks in the local NPU by that NPU's LIP. Network blocks passing through two NPUs are multiplexed twice in a remote NPU, once to/from the terminal and again from/to the local NPU.

NETWORK BLOCK HANDLING

The basic procedures for upline and downline network block movement are discussed next. Note that the procedures given are highly summarized. Acknowledgment procedures are also highly summarized. It is assumed that terminals are connected through a single NPU.

SIMPLIFIED INPUT PROCESSING

Figure 6-3 shows the movement of data from a terminal to a host application program. Solid lines indicate data and acknowledgment pathways, dashed lines indicate principal control functions.

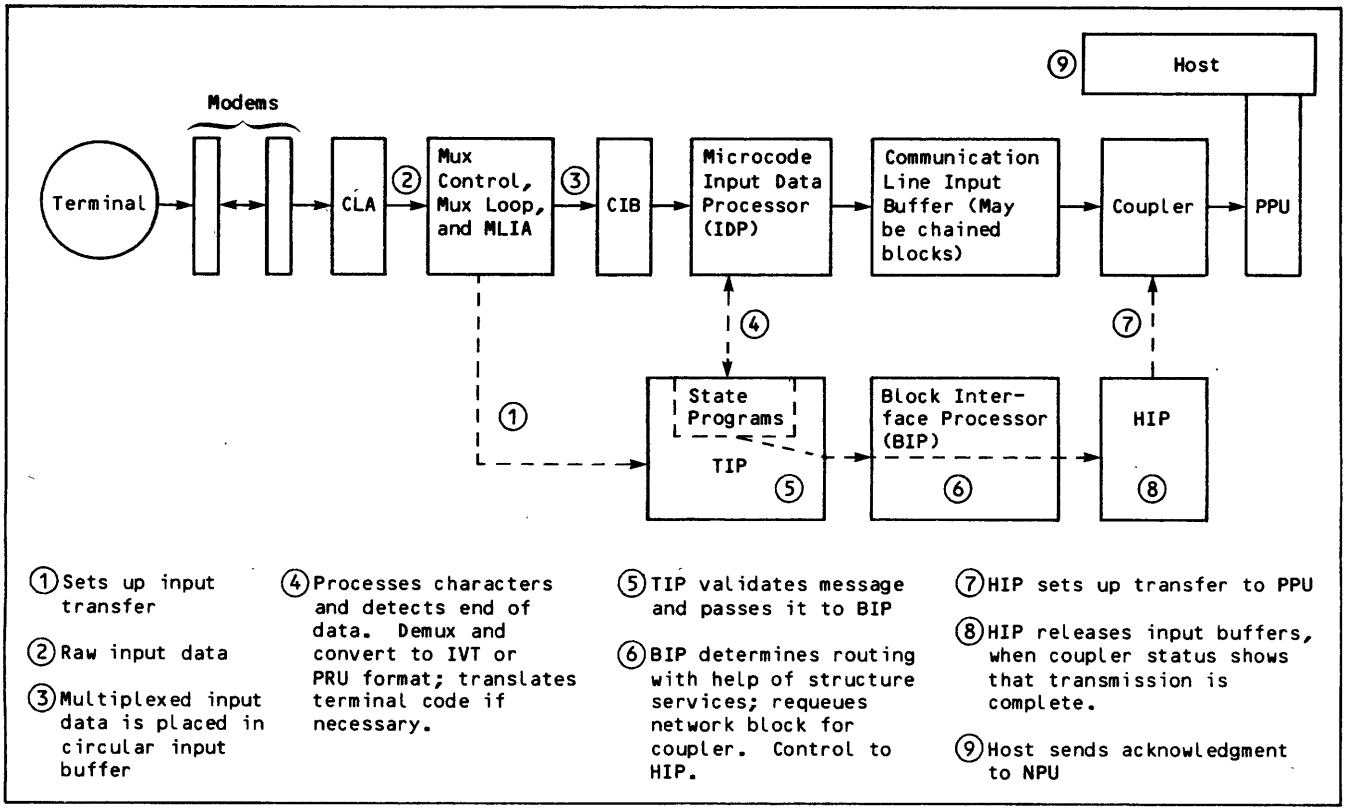


Figure 6-3. Simplified Input Processing

The major steps in upline data processing are:

Polling synchronous terminals (such as Mode 4 terminals) to find whether the terminal has data ready to send.

Setting up the multiplex subsystem and buffers when a terminal indicates it has data to send upline.

Collecting data from all active terminals in a circular input buffer (CIB).

Demultiplexing data and converting it to IVT or PRU format, a process controlled by state programs unique to each TIP. Demultiplexed data is collected in one or more chained buffers that are collectively called a communication line input buffer. If code conversion is necessary (such as EBCDIC to ASCII for IVT data or EBCDIC to display code for PRU data), this is also accomplished.

When an input buffer is full (that is, the transmission block is complete), validating the block content for conformance to the appropriate protocols.

Converting the transmission block to one or more network blocks.

Determining network block routing and queuing the block to the host coupler.

Updating line statistics for the transfer (line statistics are a type of status information that includes both success and failure data, such as number of blocks and characters processed or the number of transmission errors detected). These statistics are periodically sent upline to the host as accounting messages and hardware error log messages.

The coupler transmitting the block to the host PPU and dequeuing the block.

The host receiving the block and queuing it to the applications program.

The NPU releasing the block buffer.

The host delivering the block to an application program and sending an acknowledgment to CCP.

SIMPLIFIED OUTPUT PROCESSING

Figure 6-4 shows the movement of data from a host application program to an interactive terminal (downline blocks). Conventions for solid and dashed lines are the same as for the input diagram.

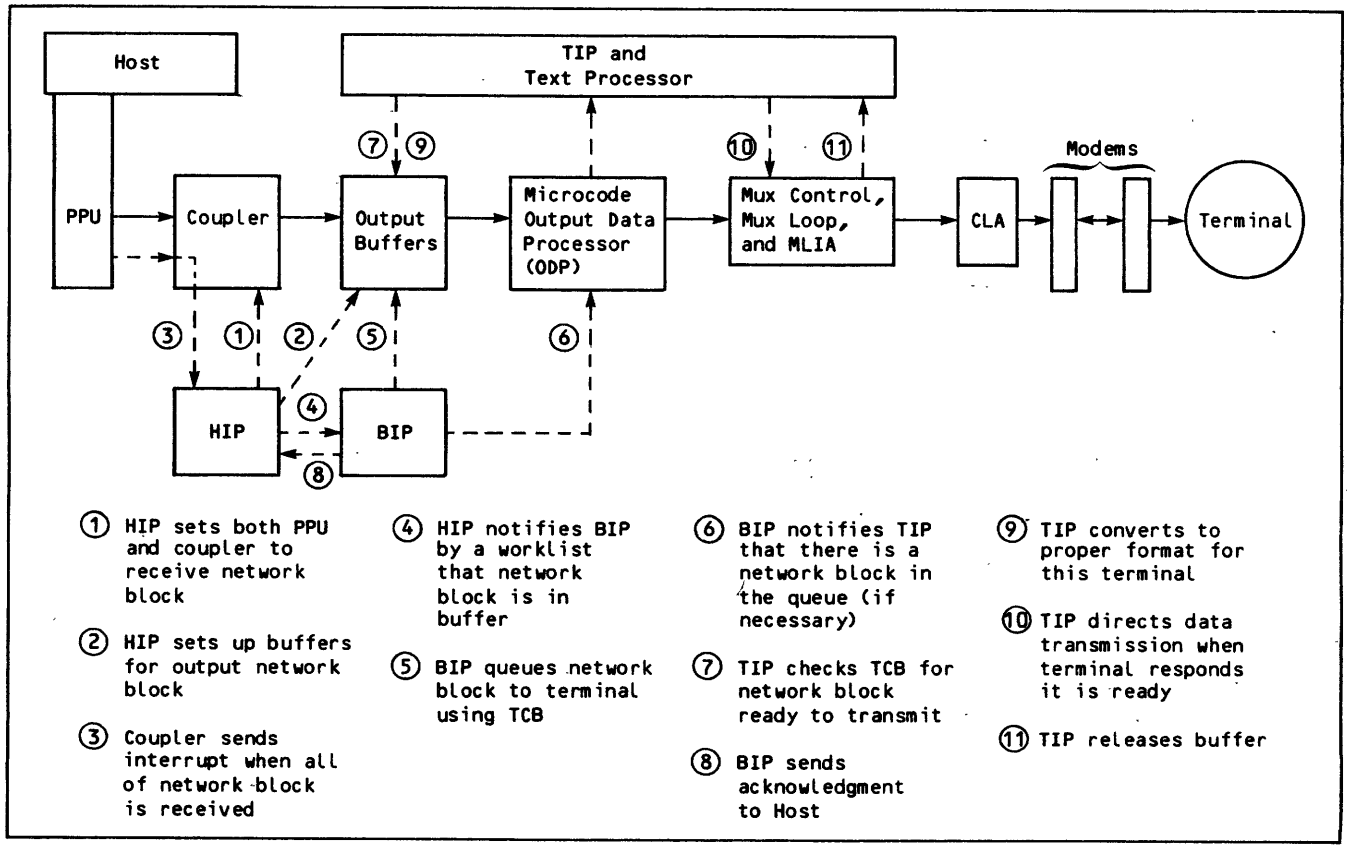


Figure 6-4. Simplified Output Processing

The text below describes the same movement for data sent to a CDC-defined batch device; the same steps occur for both device types, but the order of operations varies. The major steps in downline data processing are:

An interrupt from the host indicates a buffer of data is ready for transmission. The data is in PRU or IVT format at this point.

If the NPU is not already saturated with other, higher priority tasks (note that output takes precedence over input), the host interface package (HIP) sets up the coupler to receive the network block and assigns a buffer (or chained buffers) of space to be used as an output buffer for the block.

The network block is sent by a buffered transfer from the PPU through the coupler to the assigned buffers. The coupler causes the transmission-complete interrupt to the HIP when all of the network block has been received.

The text processor converts the network block from IVT or PRU format to the destination terminal's format (transmission blocks).

The transmission block is queued to a terminal control block where the terminal interface package (TIP) detects that data is available.

If the terminal is able to receive output data, the TIP directs the multiplex subsystem to output the message.

When a synchronous terminal has detected the end of a transmission, it sends an acknowledgment.

The NPU sends an acknowledgment to the host.

When the TIP knows that the data was received, the NPU terminates the output operation by releasing the output buffers.

Line statistics are updated for the transfer (line statistics are a type of status information that includes both success and failure data, such as number of blocks and characters processed or the number of transmission errors detected). These statistics are periodically sent upline to the host as accounting messages and hardware error log messages.

DATA PRIORITIES

Network blocks are defined to be high or low priority as a function of the device that sends or receives the data in the block. Regulation of data is needed because all network blocks require buffer space; combined peak load demands from terminals, from the host, and from neighboring NPUs (if any) might require more buffers than the NPU can assign.

To prevent NPU stoppage for lack of buffers, the NPU is allowed to reject input on the basis of priority or because the channel seeking to input data already has its maximum share of buffers assigned. For terminals that are polled for input, regulation is handled indirectly; the NPU does not poll the terminals until more buffers are available. As current network blocks are processed and output, more buffers become available and the regulation level can change to accept data previously rejected.

Two types of regulation exist:

Regulation of upline and downline data for specific connections, caused by a device using more data buffers or network blocks than it is allowed.

Regulation of upline and downline data for entire logical links (data paths through an NPU to a host), caused by a shortage of buffers in either the NPU or a host.

Connection Regulation

Data to or from a terminal device is regulated by the upline and downline network block limits assigned for the device. The host queues the number of input blocks specified as the upline block limit for the device. The NPU queues the number of downline blocks specified as the device's downline block limit. The downline queuing done by the host is established by two queuing limits, the application block limit and the downline block limit.

The application, downline, and upline block limits are initially established by the network administrator in the network definition file, using the Network Definition Language statement ABL, DBL, and UBL parameters. This process is discussed in section 2.

When the upline block limit for a device is reached, input regulation begins for the device. Input regulation is discussed separately for each TIP, later in this section. When the application block limit for downline data is reached, the host performs output regulation without involving the NPU.

Levels of Logical Link Regulation

Three levels of traffic priority are defined on a logical link:

Service message traffic (priority 1): This level includes command, reply, and status information exchanged between neighboring NPUs and between NPUs and a host. Traffic at this level is suspended only after suspension of traffic at levels 2 and 3 fails to produce the needed buffer space.

High-level (priority 2): This level is intended for data associated with an interactive type of device. The size of each network block is normally small, but the network block is quickly processed so that no processing delay is visible to the terminal user. Traffic at this level is suspended only after suspension of traffic at level 3 fails to produce the needed buffer space.

Low-level (priority 3): This level is intended for data associated with batch-type devices. The size of the network block is large (sometimes a thousand bytes or more), but since no operator interaction is involved, a small delay in network block processing is acceptable. Traffic at this level is suspended first when regulation must occur to produce the needed buffer space.

The actual assignment of priorities (levels 2 and 3) to devices occurs in the network configuration file.

When the logical link used by a device becomes subject by a host to traffic suspension at the indicated priority level, input regulation begins for the device. Input regulation is discussed separately for each TIP, later in this section.

When the NPU becomes low on buffer space, it regulates all logical links passing through it. Thereafter, downline regulation occurs through host processes or neighboring NPUs, without involving the NPU. The NPU can reject downline data when buffer regulation is in effect.

LINE STATUS

CCP updates the status of enabled communication lines. If failure of an active line is detected, the line's status is changed to down and the network operator is informed. CCP then automatically enables the line so that it can be reused without network operator intervention after the failure is corrected.

TERMINAL INTERFACE PROGRAMS

There are two basic categories of Terminal Interface Programs:

Site-written TIPs

CDC-written TIPs

A site-written TIP can consist of:

Code written specifically to support a communication protocol not currently supported by a CDC-written TIP

Code added to a CDC-written TIP to support terminals that use CDC-supported protocols but have characteristics not compatible with any CDC-defined terminal class or with any CDC-defined device type

The functions and characteristics of site-written TIPs are unknown. The configurable features of communication lines, terminals, and devices supported by such TIPs are described in the Network Access Method Version 1 Network Definition Language Reference Manual described in the preface.

The released version of CCP has separate TIPs to support the following communication protocols:

The ASYNC TIP, supporting teletypewriter-compatible or IBM-2741-compatible devices

The X.25 TIP, supporting packet-switching network terminals using a packet assembly/disassembly (PAD) facility and supporting application-to-application connections through packet-switching networks or a direct link to another NPU

The MODE4 TIP, supporting terminals and devices that use either the mode 4A or mode 4C variants of the CDC Mode 4 communication protocol

The HASP TIP, supporting terminals and devices that use the Houston Automatic Spooling Program (HASP) protocol

The BSC TIP, supporting terminals and devices that use the IBM 2780 or IBM 3780 variants of the IBM bisynchronous protocol

The 3270 TIP, supporting terminals and devices that use the IBM 3270 variants of the IBM bisynchronous protocol

ASYNC TIP

The ASYNC TIP supports terminal devices connected through dedicated or dial-up asynchronous lines. Such devices normally generate IVT data to exchange in an interactive mode with a host application program. The TIP supports one terminal device per line and expects each device to have an input and an output mechanism.

The following CDC-defined terminal classes are supported:

Terminal Class	Archetype Terminal
1	Teletype Models 33, 35, 37, 38
2	CDC 713, 722-10, 751, 752, 756
3	CDC 721
4	IBM 2741
5	Teletype Model 40-2
6	Hazeltine 2000
7	Digital Equipment Corp. VT100 (ANSI X3.64)
8	Tektronix 4014, 4114

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the ASYNC TIP.

Each CDC-defined terminal class has a default set of terminal characteristics as shown in section 5. These terminal characteristics can be changed from their defaults initially for each device through the network configuration file or subsequently by the terminal user and/or the application program through the use of commands.

The valid values and default parameters for commands in each of these terminal classes are shown in section 5. When a valid value is entered from a terminal, the TIP responds with the message:

cd ACCEPTED..

where cd is the command mnemonic. When an invalid value is entered, the TIP responds with the message:

cd reason

and the command represented by cd is ignored. The field identified as reason contains a message explaining the reason the command was ignored. See appendix B for specific messages.

The ASYNC TIP supports all IVT interface commands and features except the following ones for 2741 terminals:

Break condition as user break or cancel input (the BR command)

Paper tape input or output (the IN command PT and XP options, the OP command PT option)

Input and output regulation of or by devices, called device control (the IC and OC commands)

Input cursor positioning response (the CP command)

Block mode operation

Changing the end-of-line character code to a value other than the default delimiter (the EL command X option)

Line mode cursor positioning (the EL command CP option)

Changing parity processing (the PA command)

Transparent mode timeout or character count delimiter, or changing the character code used as a delimiter (the DL or the XL command TO or C option, or the X and Y options)

Echoplexing (the EP command)

Protocol Assumptions

This subsection describes those aspects of TIP operation that depend upon hardware use or are implementor-determined portions of asynchronous communication protocols. These aspects include:

Line access

Line speed and stop bit use

Operating modes

Point-to-Point and Multidrop Line Access

The ASYNC TIP supports point-to-point communication line access for nonaddressable terminals. Multidrop terminal configurations are not supported.

Speeds and Stop Bits

Teletypewriter-compatible terminals (classes 1 through 3 and 5 through 8) are assumed to use 2 stop bits when operating at 110 b/s and 1 stop bit when operating at speeds from 150 to 9600 b/s. IBM 2741-compatible terminals (class 4) that operate at speeds of 134.5 b/s use 1.5 stop bits; 2741-compatible terminals that operate at speeds of 300 b/s use 1 stop bit.

Modes of Operation

The logical mode of operation of the TIP is normally half-duplex, with input given priority over output. This means that output is not started when input is active. If input occurs during output, output is stopped within a few characters, and continues with the first unsent character after input completes. The TIP attempts to deliver output whenever such data is available, unless:

Input is in process

A page wait condition is in force

The only exception to this half-duplex mode of operation occurs when full-duplex operation is required by the application program. Full-duplex operation, supporting simultaneous input and output, is functional only when transparent keyboard input is selected (refer to the XK or X options of the IN command in section 5). When the TIP operates in full-duplex mode, the following features are not effective options for the terminal user or the application program:

Device input and output control (IC and OC commands)

Page waiting (PG command)

Echoplexing (EP command)

Supported Input and Output Mechanisms

The TIP supports a range of input and output mechanisms; only one input and one output mechanism can be active at any given time. Control of multiple output mechanisms (for example, a display and a hardcopy printer) is left up to the terminal.

When normalized mode input is in effect, the TIP supports the following input modes and mechanisms (refer to the IN command in section 5):

Keyboard in line mode. Input is forwarded after an end-of-line is recognized; available output is sent after the end-of-line character. The ATTN key must be pressed at a 2741-compatible terminal before the cancel-input, network control, abort-block, user-break-1, or user-break-2 function characters are entered; these characters are established through the CN, CT, AB, B1, and B2 commands described in section 5.

Keyboard in block mode (not supported for 2741 devices). Input is forwarded after an end-of-line or end-of-block character is recognized; available output is only sent after the end-of-block character. Unless the terminal is operating in full-ASCII mode (Y option of the FA command, described in section 5), an end-of-block character immediately following an end-of-line character does not cause the TIP to send an empty line to the host.

Paper tape in block mode with a fixed end-of-block indicator sequence (not supported for 2741 devices). After the last message block of queued output is delivered, the TIP sends an ASCII DC1 (X-ON) character code to start the tape reader. Input is forwarded after the end-of-line character is recognized; available output is only sent if the end-of-line character is followed by the ASCII DC3 (XOFF) character. The sequence of end-of-line and DC3 codes forms the fixed end-of-block indicator. The DC3 can be preceded by an ASCII LF character as well; the LF is not part of the forwarded input.

When normalized mode output is to be sent, the TIP supports the following output devices (refer to the OP command in section 5):

Display. Line folding and scrolling are assumed to be performed by the output device when page width (see PW command in section 5) is reached.

Printer. Line folding is performed by the TIP when page width is reached. Paging is performed by the TIP when page length is reached.

Paper tape (not supported for 2741 devices). This output mechanism is treated the same as a printer, with the exception that an ASCII DC3 character and three ASCII NUL characters are sent after the completion of every downline post-print format effector action.

In transparent mode, the keyboard or paper tape are allowed input mechanisms and the display, printer, or paper tape are allowed output mechanisms. Forwarding of input is determined by the options selected for the DL or XL commands (see section 5). Output formatting is left up to the host application program. In transparent mode, the application program is responsible for all data interpretation, including control characters.

Terminal Code Sets and Parity

The ASYNC TIP supports teletypewriter-compatible terminal devices operating with user selectable parity set to even, odd, zero, none, or ignore. The initial parity of the device is established in the network configuration file but can be changed by the application program and/or the terminal user; refer to the PA command in section 5. Teletypewriter-compatible devices use the ASCII, ASCII bit-paired APL, or ASCII typewriter-paired-APL code sets.

The TIP also supports IBM 2741-compatible terminal devices with a fixed parity (set to odd). These devices use the EBCD, EBCD-APL, Correspondence Code, or Correspondence-Code-APL code sets.

The TIP converts terminal code sets to and from network ASCII codes for IBM 2741-compatible devices. The terminal code set used can be initially established in the network configuration file. The terminal user and/or the application program can change the code set as part of the automatic recognition process, using the AR command described in section 5.

Normalized Mode Output

The host sends 7-bit ASCII character codes. The eighth bit (bit 7) can be anything and is ignored. If translation is needed, bit 7 is set to zero before the TIP translates the character. If translation is not needed, the bit is set to zero before going to the CLA for output. The CLA provides the parity setting appropriate for the parity option selected. This is true for all parity settings.

Transparent Mode Output

The host can send anything for bit 7 of all parity settings; the host value for bit 7 is ignored for all parity options except parity of none or ignore. For parity of none or ignore, the entire 8 bits, as sent from the host, are sent through the CLA to the terminal. For other parity settings, bit 7 is set to zero by the TIP and sent to the CLA for output. The CLA provides the parity setting appropriate for the parity option selected.

Normalized Mode Input

The TIP sends the host 7-bit ASCII character codes with the eighth bit always set to 0. This is true for all parity settings.

Transparent Mode Input

For parity of none or ignore, the host receives the 8 bits just as they came in on the line. For the other three parity settings, the eighth bit is set to zero.

Initial Connection

The initial line speed and code set can be explicitly set through the network configuration file or can be determined by the TIP through automatic recognition. Recognition of line speed is available in a range of 110 to 9600 b/s.

Automatic recognition of line speed is performed when a line becomes active. The TIP sets the CLA to sample the line at 800 b/s. The user has 60 seconds to enter input. The first input on the line is presumed to be a carriage return code; the bit pattern received is compared against the patterns expected for that code at supported speeds until a match is found. The user then enters more input. If the next character entered is also a carriage return, the terminal is assumed to use:

The ASCII code and character set if the terminal is recognized as other than an IBM 2741

The EBCD code and character set if the terminal is using a 134.5 b/s speed, or is known to be an IBM 2741 operating at 300 bits per second

If the second character entered is not a carriage return, it is presumed to be a right parenthesis code; the bit pattern received is compared against the patterns expected for that code in supported code sets, until a match is found.

The need for automatic recognition is established through the network configuration file. Once line speed and code set are known, the terminal connection to the NPU is complete. Subsequent connection to a host and an application program depend on the network configuration; the procedure and possible commands required are described in section 4.

The security of a user's login information is insured by the CCP's maintenance of a security control character (for each sub-TIP type).

The security character is site assigned (there is no default) during a CCP build. The character must be a 7-bit character within the code set of the terminal specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B

through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

After initial connection, the automatic recognition procedure can be restarted by the terminal user without disconnecting the terminal from either the host application program or the NPU. See the AR command description in section 5.

Disconnection

The ASYNC TIP disconnects a terminal on a dial-up line when:

- The terminal uses a line speed or code set not recognized by the TIP (after 60 seconds of attempted initial automatic recognition).
- The terminal user fails to complete both steps of the automatic recognition sequence within the time allowed.
- The terminal is not connected to a host after two minutes.
- The Communications Supervisor requests that the line or device be disabled.
- The user hangs up the phone.

The ASYNC TIP does not disconnect a terminal on a dedicated line. Input from disabled terminals or input on disabled hardwired lines is discarded. Failure to complete initial automatic recognition on a dedicated line causes the TIP to restart the procedure.

Input Operations

Section 2 describes the general case for any interactive virtual terminal input operation. This subsection describes the special cases and subsets of those operations unique to terminals serviced through the ASYNC TIP.

During input, the TIP detects the end of a physical line when:

- A line feed code is entered and an upline blocking factor of 0 is in use (an upline network block is generated when 0 is in use, even though the corresponding network block size is 100 characters).
- The page width is greater than 56 characters and the page width is reached.
- The page width is 0 or greater than 56 characters and an end-of-line or end-of-block indicator is entered.

The TIP does not discard stored characters of a physical line in response to backspace codes when the backspaces occur after the end of the physical line. The user cannot backspace across a physical line boundary to delete previously entered characters.

When input ends, the TIP can perform cursor positioning as a response to that condition. Whether cursor positioning occurs is determined by the current setting of the CP command (refer to section 5). The response used is determined by the current value for the CP option of the EL or EB command. For 2741 terminals, cursor positioning always occurs after input because the terminal ends each line with a newline code (NL).

The following operations can be correctly supported by the ASYNC TIP only in line mode:

- Cursor positioning responses at the end of each physical and logical line (refer to the Y option of the CP command in section 5); cursor positioning at the end of each block is the only option that can be supported in block mode.
- Special editing mode processing of the backspace-linefeed code sequence.
- Input flow control (refer to the IC command in section 5), this does not work for devices operating in block mode.

For 2741 terminals, the TIP keeps track of shifts between uppercase and lowercase. Each physical line of input is presumed to start in lowercase.

For paper tape operation (refer to the PT option of the IN command in section 5), a DC3 (X-OFF) code is sent upline as data unless it follows an end-of-line indicator. For transparent paper tape operation (refer to the XP option of the IN command), a DC3 (X-OFF) code following an end-of-line indicator and the end-of-line indicator itself are sent upline as data when that sequence does not delimit multiple message transparent input.

The three editing modes described in section 2 are supported for all terminal classes.

Output Operations

Three interactive virtual terminal output control operations are unique to the ASYNC TIP:

- Discarding output on terminal user or application program command
- Suspending output in response to device commands that perform output regulation
- Processing of physical line endings

Discarding Output

If the terminal user enters the abort output block character (refer to the AB command in section 5), the ASYNC TIP discards the rest of the current transmission block and the current network block. Output continues with any subsequent block received from the host.

Output Regulation

Terminals can transmit an isolated ASCII DC3 (X-OFF) code to stop output. When output control by a device is selected (refer to the Y option of the OC command in section 5), the TIP stops output within n characters (n depends on the system load, but is normally 2) and discards the received DC3 code.

The terminal can then transmit an isolated ASCII DC1 (X-ON) code to resume output. When the TIP receives that code, it discards the DC1 and resumes output with the next character in the current buffer.

If the terminal does not transmit an ASCII DC1 (X-ON) code, any input followed by an end-of-block or end-of-line indicator causes output to resume. The TIP discards the end-of-block or end-of-line indicator if appropriate for the current operating mode, and the received input is sent to the host as data.

The ASYNC TIP also supports output regulation through use of the clear-to-send (CTS) modem signal. The TIP stops output within two characters when CTS drops and resumes output when CTS is raised again.

Physical Lines

During output to a device with a printer mechanism (refer to the PR option of the OP command in section 5), a physical line ends when the page width is reached or when the output contains a unit separator code, a carriage return code, or a line feed code. During output to a device with a display mechanism (DI option of the OP command), a physical line ends when the output ends or when it contains a unit separator code, a carriage return code, or a line feed code.

For devices with printer mechanisms, the TIP inserts carriage return, line feed, and idle codes when needed after the end of a physical line; refer to the PW, CI, and LI commands in section 5.

Break Significance

A break condition on an asynchronous communication line exists when a null character with a framing error is detected by the CLA. Such a condition causes the TIP to stop any output in progress and wait for input. This condition has the following additional meaning to the ASYNC TIP only when normalized mode input is used (IN command KB or PT or BK options, described in section 5).

A device can be defined such that detection of a break condition on the line is interpreted as either a user-break-1 signal (refer to the B1 command in section 5) or as a cancel-input signal (refer to the CN command in section 5). This is done initially through the network configuration file and subsequently by an application program and/or a terminal user through the Y option of the BR command, described in section 5.

If input is in process when such a break condition is detected, the condition is treated as a request to cancel input. A flag is set, and the canceled data is sent upline to the host for discarding. The set flag becomes the can bit in the application block header. As it does when it detects the cancel character at the end of a line, the TIP sends the message

```
*DEL*
```

to the terminal.

If the terminal is idle or output is in process when such a break condition is detected, the condition is treated as a user-break-1 request. An interrupt and break indicator block sequence is created, and all data queued for output is discarded; the block sequence is sent upline to the host. In the host, the sequence of blocks becomes a supervisory message.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input. For devices that support input control (refer to the Y option of the IC command in section 5), input is stopped whenever a temporary suspension of input must occur. The TIP sends an ASCII DC3 (X-OFF) code to accomplish this. When regulation ends, the TIP sends an ASCII DC1 (X-ON) code to these terminals.

If input of a logical line occurs when regulation is needed, the message

```
WAIT..
```

is sent. The data transmitted from the terminal is discarded. For terminals operating in block mode, a varying number of logical lines within a block might be discarded; that is, regulation can begin after the first logical line is stored and can end before the last logical line of the transmission block begins. The WAIT message is sent for each discarded line.

When data has been discarded, the message

```
REPEAT..
```

is sent as soon as regulation ends and data can again be accepted and stored by the NPU. Paper tape readers must be restarted manually.

Output from the host is sent to the terminal during input regulation, unless input was in process when regulation began.

If regulation begins after input has been accepted by the Block Interface Program but before the block can be queued for transmission to the host, the message

```
INPUT DISCARDED..
```

is sent to the terminal and the block is discarded.

The ASYNC TIP also supports input regulation through use of the request-to-send (RTS) RS232 modem interface.

Error Recovery

The ASYNC TIP performs no error recovery on input data; any error recovery needed must be performed by the application program in the host. If a parity error or framing error is detected during input, the TIP stores the erroneous code in the network block and sets a flag. The flag becomes the pef bit in the host application block header.

A line failure occurs whenever the TIP detects a carrier signal drop lasting more than one second, or any drop in the data set ready signal. The TIP performs no recovery for line failures. The terminal must reconnect to the network.

X.25 TIP

The X.25 TIP PAD subTIP supports asynchronous terminal devices connected through a packet-switching network (PSN) or a public data network (PDN). These networks provide a packet assembly/disassembly (PAD) service with separate switched virtual circuits for each device, and use a dedicated synchronous communication line to exchange data with the NPU. The terminal device is connected to the PDN or PSN over a voice grade line at low or medium speeds to the PAD service.

X.25 devices normally exchange IVT data with a host application program. The TIP supports one terminal device per virtual circuit and expects each device to have an input and an output mechanism.

The X.25 TIP application-to-application subTIP (called X25) supports host application programs connected through a packet-switching network or a trunk between NPUs. Each connection uses a separate virtual circuit.

Application-to-application data is always transparent to the network software.

The following CDC-defined terminal classes are supported:

Terminal Class	Archetype Terminal
1	Teletype Models 33, 35, 37, 38
2	CDC 713, 722-10, 751, 752, 756
3	CDC 721
5	Teletype Model 40-2
6	Hazeltine 2000
7	Digital Equipment Corp. VT100 (ANSI X3.64)
8	Tektronix 4014, 4114

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the X.25 TIP PAD subTIP.

Each CDC-defined terminal class connected via a PAD service has a default set of terminal characteristics, as shown in section 5. Some of these characteristics differ from those the terminal class would possess if serviced through the ASYNC TIP. These terminal characteristics can be changed initially for each communication line through the network configuration file or subsequently by the terminal user and/or the application program through the use of commands.

The valid values and default parameters for commands in each of these terminal classes are shown in section 5. When a valid value is entered from a terminal, the TIP responds with the message:

cd ACCEPTED..

where cd is the command mnemonic.

When an invalid value is entered, the TIP responds with the message:

cd reason

and the command represented by cd is ignored. The field identified as reason contains a message explaining the reason the command was ignored. Appendix B describes these messages.

The X.25 PAD subTIP supports all IVT interface commands and features except the following:

Paper tape input or output (the IN command PT and XP options, the OP command PT option).

Abort output blocks (the AB command)

Cancel input using a break condition key (the BR command).

Transparent mode timeout delimiter (the DL or XL command TO option).

Protocol Assumptions

This subsection describes those aspects of TIP operation that depend upon hardware use or are implementor-interpreted portions of the X.25 communication protocol. These aspects include:

Frame and packet characteristics

Operating mode

Frame and Packet Characteristics

A PSN or PDN uses the CCITT X.25 protocol to transfer data over a high-speed communication line called a link. A PSN or PDN can have one or more links to a single NPU. An NPU can service more than one PSN or PDN.

Each link from a PSN or PDN can carry one or more virtual circuits. Virtual circuits can be either switched or permanent. Terminals using a PAD service always use switched virtual circuits; application-to-application connections always use switched virtual circuits. Each terminal device or application-to-application connection using the link uses a separate virtual circuit on the same physical communication line; the X.25 TIP multiplexes the virtual circuits as a software function.

The PAD service protocol is defined by the following CCITT recommendations:

Recommendation X.3 defines the conversion of asynchronous character-by-character ASCII data to a synchronous packet format. This recommendation provides options that govern PAD operation; these options are selected when the user subscribes to the PDN or PSN.

Recommendation X.28 defines PAD options for the terminal.

Recommendation X.29 defines PAD options for the NPU.

Because some of these selections affect network operation, the PAD subTIP sets some of the standard PAD reference selections common to the X.25 networks listed in the Network Access Method Version 1 Network Definition Language Reference Manual (see preface).

Upline, the PAD service consolidates terminal data into packets, adding prefix and suffix information when necessary. Packets are placed in frames by the PDN and transmitted over the link to the NPU.

The TIP discards all frame and packet prefix and suffix information and forwards the terminal data to the host as network blocks of the appropriate block type, depending on the upline block size and the presence of end-of-line or end-of-block indicators. The end-of-packet sequence bit (an M bit value of 0) is interpreted as an end-of-block indicator.

Downline, the TIP collects data into packets. Each network block becomes one or more packets; the continuation bit (the M bit) of the packet is set to one in all packets except the last created from the last network block of a logical line (the end of a message); messages continued across network blocks are marked as continuation packets. The qualified data bit (the Q bit) is set to one in all packets where that is requested by the sending application program for the corresponding block. The TIP then assembles the packets into frames for transmission over the link. Each frame contains data for only one virtual circuit.

The PAD service disassembles packets into characters of data for a terminal.

Mode of Operation

Both subTIPs are insensitive to line speeds. The logical mode of operation of both subTIPs is half-duplex. The PAD subTIP delivers output whenever such data is available, unless:

Input is in progress

A page wait condition is in force

The X25 subTIP delivers output data whenever such data is available, unless:

Input is in progress

A reset request is outstanding

There is no exception to this half-duplex mode of operation. When full-duplex operation is requested by the application program, the subTIP cannot honor the request.

Supported Input and Output Mechanisms

The PAD subTIP supports one input and several output mechanisms; only one input and one output mechanism can be active at any given time. Control of multiple output mechanisms (for example, a display and a hardcopy printer) is left up to the terminal.

When normalized mode input is in effect, the PAD subTIP supports only one input mode and mechanism (refer to the IN command in section 5), the keyboard in block mode. Input is forwarded after an end-of-line or end-of-block indicator is recognized (the end-of-packet continuation bit, bit M, is always an end-of-block indicator). If an end-of-block indicator in addition to the M bit is defined, that character is discarded. Available output is only sent after an end-of-block indicator is received.

When normalized mode output is to be sent, the PAD subTIP supports the following output mechanisms (refer to the OP command in section 5):

Display. Line folding and scrolling is assumed to be performed by the output devices when page width is reached (see the PW command in section 5).

Printer. Line folding is performed by the TIP when page width is reached. Paging is performed by the TIP when page length is reached (see the PL command in section 5).

In transparent mode, the keyboard is the allowed input mechanism and the display or printer are allowed output mechanisms. Forwarding of input is determined by the options selected for the DL or XL commands (see section 5). Output formatting is left up to the host application program. In transparent mode, the application program is responsible for all data interpretation, including control characters.

Code Sets and Parity

The PAD subTIP supports teletypewriter-compatible terminal devices operating with user-selectable parity set to even, odd, zero, none, or ignore. The initial parity of the device is established in the network definition file but can be changed by the application program and/or the terminal user; refer to the PA command in section 5. Devices accessing the NPU through PAD services can use the ASCII, ASCII typewriter-pairing APL, or ASCII bit-pairing APL code set.

The X25 subTIP supports binary transparent mode data with no code set association. A parity setting of none is always used for application-to-application connections.

Normalized Mode Output

The host sends 8-bit ASCII character codes. The eighth bit (bit 7) can be anything and is ignored. Bit 7 of the host output is replaced before going to the CLA for output; the TIP supplies a value for bit 7 that is appropriate for the selected parity option. This is true for all parity settings.

Transparent Mode Output

The host can send anything for bit 7 of all parity settings; the host value supplied for bit 7 is ignored for all parity options except a parity of none or ignore. For parity of none or ignore, the entire 8 bits, as sent from the host, are sent through the CLA to the terminal. For other parity settings, bit 7 is set to zero by the TIP and sent to the CLA for output. The CLA provides the parity setting appropriate for the parity option selected.

Normalized Mode Input

The TIP sends the host 8-bit ASCII character codes with the eighth bit always set to 0. This is true for all parity settings.

Transparent Mode Input

For parity of none or ignore, the host receives the 8 bits just as they came in on the line. For the other three parity settings, the eighth bit is set to zero. Parity is not checked for correctness; for terminals, parity correctness is determined by the PAD service before the data is received by the NPU.

Initial Connection

Automatic recognition through the TIP is supported. Refer to the AR command in section 5.

The X.25 TIP supports either data terminal equipment (DTE) or data channel equipment (DCE) termination protocol. The X.25 TIP can also accept service charges for incoming calls. Both aspects of virtual circuit initialization must be known before terminal connection to the NPU can be completed.

Protocol terminating location information and service charge acceptance requests are supplied by the PSN or the PDN and must agree with information supplied by the network administrator in the network configuration file.

When terminal connection to the NPU is complete, subsequent connection to a host and an application program occur. That connection process depends on the network configuration; the procedure and possible commands required are described in section 4.

The security of a user's login information is ensured by the CCP's maintenance of a security control character (for each subTIP type).

The security character is site assigned (there is no default) during a CCP build. The character must be a 7-bit character within the code set of the terminal specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

Disconnection

The X.25 TIP clears a switched virtual circuit when the Communications Supervisor disables a terminal, a terminal disconnect occurs, an application-to-application connection ends, or a terminal is not connected to a host for two minutes.

Input Operations

Section 2 describes the general case for any interactive virtual terminal input operation. This subsection describes the special cases and subsets of those operations unique to terminals serviced through the X.25 TIP.

During input, the PAD subTIP detects the end of a physical line when:

A line feed code is entered and an upline blocking factor of 0 is in use (an upline network block is generated when 0 is in use, even though the corresponding network block size is 100 characters).

The page width is greater than 56 characters and the page width is reached.

The page width is 0 or greater than 56 characters and an end-of-line or end-of-block indicator is entered.

The PAD subTIP does not discard stored characters of a physical line in response to backspace codes when the backspaces occur after the end of the physical line. The user cannot backspace across a physical line boundary to delete previously entered characters.

When input ends, the TIP can perform cursor positioning as a response to that condition. Whether cursor positioning occurs is determined by the current setting of the CP command (refer to section 5). The response used is determined by the current value for the CP option of the EL or EB command.

The following operations can be correctly supported by the X.25 TIP only in line mode:

Cursor positioning responses at the end of each physical and logical line (refer to the Y option of the CP command in section 5); cursor positioning at the end of each block is the only option that can be supported in block mode.

Special editing mode processing of the backspace-linefeed code sequence.

The user of an asynchronous terminal addresses an X.25 packet-switched network through a PAD.

You should be aware of two areas of interaction between PAD parameters and the NPU PAD subTIP parameters. The first area is that the PAD forwarding parameter needs to be specified as identical with either the network's end-of-logical-line or its end-of-block character (or both). The default (end-of-logical-line) is a carriage return which corresponds to PAD parameter number 3 with a value of 2.

Unless one of these conditions prevails, the network software passes the PAD forwarding character as data to the application program. Conversely, if the forwarding signal chosen corresponds with either end-of-block or end-of-logical-line, then it is removed from the stream delivered to the application program.

Multiple logical lines (each terminated with a [cr]) can be packed into a single packet, if you use the

ct EL [cr]

terminal definition command to set the EL character to [cr] and the PAD forwarding signal to an EOT or ETX character.

Setting the EL, EB, DL, or XL characters using terminal definition commands is restricted to the control characters defined in columns 0 and 1 of the ASCII character set (see appendix A).

Output Operations

During output to a device with a printer mechanism (refer to the PR option of the OP command in section 5), a physical line ends when the page width is reached or when the output contains a unit separator code, a carriage return code, or line feed code. During output to a device with a display mechanism (DI option of the OP command), a physical line ends when the output ends or when it contains a unit separator code, a carriage return code, or a line feed code.

For printer mechanisms, the PAD subTIP inserts carriage return, line feed, and idle codes when necessary after the end of a physical line; refer to the PW, CI, and LI commands in section 5.

Some PAD services recognize an ASCII DC3 (X-OFF) code from a terminal as a control signal to interrupt output. A subsequent ASCII DC1 (X-ON) code serves to resume output from the PAD. If the PAD service does not support output control, any DC3 (X-OFF) or DC1 (X-ON) code sent by the terminal is received by the TIP, treated as data, and sent upline to the host application program.

NPU cursor positioning can be turned off (it is turned on, by default) with the

```
ct CP = N [cr]
```

terminal definition command. This saves a packet on each line (or block, if you are running in block mode).

Break and Reset Significance

A break condition on an X.25 communication line is a selectable option of a PAD service. A device can be defined such that occurrence of a break condition on the line is interpreted as a user-break-1 signal (refer to the B1 command in section 5). This is done initially through the network configuration file and subsequently by an application program and/or a terminal user through the Y option of the BR command, described in section 5.

When a break occurs, the TIP discards all data queued for output to the PSN or PDN. The TIP creates an interrupt and break indicator block sequence and sends it upline to the host. In the host, the sequence blocks become supervisory messages.

After the host indicates that output can resume, the TIP retains subsequently received downline data until input from the terminal occurs. The PAD subTIP notifies the PAD service so that subsequent input is allowed.

When a break occurs, the PAD service discards all downline data queued from the TIP but not received by the terminal. The PAD might not discard upline data queued from the terminal within the PDN or PSN.

Because the X.25 protocol does not allow the TIP to determine whether input was in process when the break condition occurred, the TIP cannot interpret a break condition as a request to cancel input.

For application-to-application connections, the X25 subTIP detects reset packets and forwards them to the host (where they become supervisory messages). Reset supervisory messages from the host become reset packets in the output. Similarly, the host can send or receive interrupt supervisory messages; the X25 subTIP processes such messages as interrupt packets.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input. When regulation is needed, the TIP stops acknowledging incoming packets. Unacknowledged packets are retained and are acknowledged once regulation ends. At some point while regulation is occurring, the PDN or PSN will stop sending packets because the packet window (the permitted number of unacknowledged packets) is reached.

The terminal user or sending application is not informed by the TIP when regulation is in process. Packets that cannot be sent to the TIP from the PDN or PSN are presumed to be queued by the PDN or PSN so that no data is lost.

Output from the host is sent to the terminal or the other application during input regulation, unless input was in process when regulation began.

Some PDN PAD services support DC3 (X-OFF) and DC1 (X-ON) codes as device input control codes (refer to the Y option of the IC command in section 5).

Error Recovery

Because the TIP does not check parity on input, parity errors are not detected or reported to the host for corrective action.

At the packet level, the TIP checks for missing input packets. A missing packet causes a channel reset. Other packets traveling upline for the virtual circuit are discarded; no action is taken for packets going downline.

The action taken by the PDN or PSN in response to a channel reset depends on the specific PDN or PSN. If the PDN or PSN does not retransmit the packet, the data is lost. The TIP does not inform the terminal user of a channel reset.

If the TIP receives a reset packet from the PDN, it reinitializes the channel in both directions. All packets traveling upline and downline on the same circuit are discarded. The TIP does not retransmit packets in response to a reset packet and does not inform the terminal user that upline or downline data was discarded. In the current release, the TIP does not notify the host application program of a channel reset for either cause on a device-to-application connection. On an application-to-application connection, the upline application is notified that a reset is needed.

If the TIP receives a restart packet from the PDN, it reinitializes the circuits in both directions. All packets travelling upline and downline on the

same circuit are discarded. The TIP does not retransmit packets in response to a restart packet and does not inform the terminal user that upline or downline data was discarded. In the current release, the TIP does not notify the host application program of a circuit restart for either cause on a device-to-application connection. On an application-to-application connection, the upline application is notified that a reset is needed.

At the link level, the TIP checks for bad frame content, for missing frames, and for bad frame formats. If such an error is recoverable (the frame is identified and was retained for retransmission), the frame is resent. Otherwise, the link is reset. The action taken for a link reset depends on the specific PDN or PSN. The TIP does not retransmit downline packets after a link reset and does not notify either the host application program or the terminal user on a device-to-application connection. On an application-to-application connection, the TIP notifies the upline application that a reset is needed.

MODE4 TIP

The MODE4 TIP supports terminals using CDC Mode 4 protocol connected through dedicated or dial-up synchronous lines. Such terminals contain devices that exchange IVT data with a host application program in an interactive mode, and devices that exchange PRU data with a host application program in a batch mode.

The TIP expects each terminal to have at least one interactive device. The interactive device must have an input and an output mechanism.

The terms cluster, terminal, and device have varying definitions within mode 4 documentation. This subsection uses these definitions:

A terminal is a group of devices with a common cluster address.

A cluster is a group of devices with a single cluster controller.

A device is a separately accessible unit of input or output equipment, such as a console, a card reader, or a line printer. Within the network, a device is separately addressable, regardless of whether it has a separate hardware address within the terminal. For mode 4C terminals, devices have separate terminal addresses; for mode 4A terminals, devices use a fixed terminal address (that of the console device) and select interactive or batch access through escape codes in the data.

Cluster addresses and terminal addresses are described in the operating guide for your terminal.

A CDC-defined mode 4A terminal has one console, one card reader, and one line printer. A CDC-defined mode 4C terminal can have one or more consoles and line printers.

The following CDC-defined terminal classes are supported:

Terminal Class	Archetype Terminal
10	CDC 200 User Terminal (mode 4A)
11	CDC 714-10/20 display terminal (mode 4C)
12	CDC 711-10 display terminal or Tektronix 4014 terminal (mode 4C)
13	CDC 714-30 display terminal (mode 4C)
15	CDC 734 and 731 remote batch terminal (mode 4A)

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the MODE4 TIP.

Each CDC-defined terminal class has a default set of terminal characteristics as shown in section 5. These terminal characteristics can be changed from their defaults initially for each terminal or device through the network configuration file or subsequently by the terminal user and/or the application program through the use of commands.

The valid values and default parameters for commands in each of these terminal classes are shown in section 5. When a valid value is entered from the console, the TIP responds with the message:

```
cd ACCEPTED..
```

where cd is the command mnemonic. When an invalid value is entered, the TIP responds with the message:

```
cd reason
```

The command represented by cd is ignored. The field identified as reason contains a message explaining the reason the command was ignored. Appendix B describes these messages.

The MODE4 TIP supports all of the IVT interface commands and features except the following:

Automatic recognition restart (the AR command)

Parity selection (the PA command)

Special editing (the SE command)

Paper tape input or output (the IN command PT and XP options, and the OP command PT option)

Console printing mechanism selection (the OP command PR option)

Input and output regulation of or by devices, called device control (the IC and OC commands)

Abort output blocks (the AB command)

Cursor positioning (the EL and EB command CP option or the CP command)

Line mode operation (the IN command KB option)

Changing the end-of-block character code to values other than the default (the EL and EB command Y option)

Transparent mode timeout or character count delimiter, or changing the character code used as a delimiter (the DL or the XL command TO or C option, or the X and Y options)

Echoplexing (the EP command)

Break condition as user break or cancel input (the BR command)

Idle character insertion at the end of physical lines (the CI and LI commands)

Backspace processing (the BS command)

Protocol Assumptions

This subsection describes those aspects of TIP operation that depend upon hardware use or are implementor-interpreted portions of the CDC Mode 4 communication protocols. These aspects include:

Unsupported protocol features

Polling

Line access

Operating mode

Toggle bit use is described as part of Error Recovery, later in this section.

Unsupported Protocol Features

The MODE4 TIP does not use or support the following Mode 4 protocol message types:

Diagnostic write

Alert

Initialize command or request

Terminal disconnect

Network disconnect

Polling

The MODE4 TIP polls each mode 4A console and requests status from mode 4C clusters on a timed basis. The TIP uses a fast and slow method of service based on whether there was any recent data traffic from or to the terminal. Mode 4C consoles are polled only when the cluster controller indicates an active read request in the cluster's status response.

Point-to-Point and Multidrop Line Access

Mixtures of mode 4A and mode 4C terminals are supported on multidrop lines. Mixtures of ASCII and External BCD terminals on the same line are also supported.

Operating Mode

The MODE4 TIP is insensitive to line speeds.

The logical mode of operation of the TIP is half-duplex for each device, with console input given priority over batch output. The TIP attempts to deliver console output whenever such data is available, unless:

A page wait condition is in force

An automatic input mode block has been delivered to the terminal

The end of an input transmission block is signified by an ETX character code. This end-of-block indicator is part of the Mode 4 protocol and cannot be changed.

Supported Input and Output Mechanisms

The TIP supports several input and output devices and mechanisms. Console devices have a keyboard as an input mechanism and a display as an output mechanism.

CDC-defined mode 4A batch devices are either card readers or impact line printers. Such devices are serviced sequentially with the console.

CDC-defined mode 4C batch devices are either impact printers or nonimpact printers. Such devices are serviced concurrently with the console.

Sequential Operation

For mode 4A terminals, only one interactive or batch input or output mechanism can be active at any given time. Input and output are therefore sequential operations.

The TIP suspends console operations when either the card reader becomes active or data is available for the line printer and no console data is available. Batch input and output blocks alternate (are interleaved) until one of the following events occurs:

Output arrives for the console from the host.

The card reader or printer is stopped (or suspended).

The card reader stops and no printer data is available for 5 seconds.

Console input is attempted (the batch interrupt key is used).

In the first three cases, the TIP issues a clear write to clear the terminal buffer. In the last case, the TIP polls the console for input data.

Concurrent Operation

For mode 4C terminals, all devices can be active at a given time, but only the input or output mechanism of the console device can be active at a given time. Input and output are therefore concurrent operations for batch devices but sequential operations for the console. Interactive operations can be concurrent with batch operations.

Card Reader Operation

Card reader operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Start input

Resume input

Abort input

The TIP sends command blocks to the host to:

Report a not ready status returned by the card reader

Report an interruption in batch operation

Report incomplete card images (fewer than 80 characters)

Report incomplete input files (when the last card read is not the end of information)

Report the end of the input stream

Report the number of cards read

Report the end of a file

A mode 4 card reader becomes ready for input when the host sends the TIP a start input command block. The TIP then polls the terminal for input.

Card input begins with the first card read. A card input file begins when the beginning-of-information is detected. The TIP expects a nonblank data card as the beginning-of-information initially or following an end-of-information card. The TIP discards blank cards, end-of-record cards, or end-of-information cards read when it is expecting the beginning of information.

When the beginning-of-information is located, the TIP converts subsequent data into a NOS operating system job file in 6-bit display code. Conversion from 026 or 029 punch patterns is performed in the terminal hardware, so transparent card input (binary cards) or changes in punch translation cannot be supported.

The TIP converts the received card data from either External BCD code or ASCII code to 6-bit display code. Trailing blanks are discarded unless the last nonblank character is a colon; one blank is preserved or inserted when the last character is a colon or the card contains an odd number of significant characters. For hosts using a 64-character set, this convention produces an 82-character record when the character in column 80 is a colon.

Every card image becomes a zero-byte terminated record. These records become a network block equivalent to one, two, or three physical record units (PRUs); a block equivalent to one PRU contains 640 6-bit characters or equivalent zero-byte terminator bytes, left-justified in an 8-bit character byte. A record is continued across two network blocks when necessary. In the host, NAM removes the two low-order bits and packs the characters into actual PRUs.

The TIP recognizes /*EOR and 7/8/9 multipunch cards as end-of-record indicators for NOS system logical records. It converts any numbers found in columns 2 and 3 or 6 and 7 to octal record level numbers. When necessary, short or zero-length PRUs are created.

The TIP recognizes /*EOI and 6/7/8/9 multipunch cards as end-of-information indicators for NOS files.

Printer Operation

Line printer operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are coverted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report an interruption in batch operation (mode 4A only)

Report a printer not-ready condition (mode 4A only)

Report when a file limit is exceeded

Report when a printer message (PM) is received

Report the number of lines printed

Report the end of a file

The TIP begins output to a printer when it has output queued for the device. Data received from the host is packed as line images in PRU-sized network blocks (NAM strips zero byte terminators and replaces them with hexadecimal FF code bytes). The TIP discards trailing blanks on a print line.

Six-bit display code is converted to either External BCD or ASCII as appropriate for the terminal. ASCII print files are converted from 7-bit codes right-justified in 8-bit bytes to the codes supported for normalized mode data (see Terminal Code Sets and Parity). This conversion includes stripping the unneeded bit 7 (the host PIP routine strips the unneeded bits 8 through 11 prior to downline transmission). The parity bit is appropriately set during output.

Console input or output interrupts printing at a mode 4A terminal; the TIP reports this condition to the application program through a command block.

Mode 4C printer busy and not-ready conditions are not reported to the host, and printing continues when the condition clears.

Once started, the TIP continues output to mode 4C printers unless stopped or interrupted by a command block from the host or a printer message (PM carriage control sequence in the print file). Any mode 4C or mode 4A printer that is interrupted, stopped, or not ready can be restarted by a command block from the host.

The TIP supports both mode 4C impact and nonimpact printers. The impact printer supports page eject and can be controlled through carriage control character processing. The nonimpact printer does not support page eject, so the TIP simulates that effect when it processes carriage control characters. This simulation uses the defined page length of the device (from the network definition file) to calculate the number of lines remaining to the bottom of the page; the TIP inserts the calculated number of line feeds into the output, then adds six more to space over the simulated page boundary.

Print Message Processing

When the print file contains a line beginning with the characters PM, the TIP stops the printer, notifies the application program through a command block, and sends the data from the print file line to the application program.

Normalized Mode Console Operations

When normalized mode input is in effect from a console, the TIP supports only one input mode and mechanism (refer to the IN command in section 5): the keyboard in block mode. Input is forwarded after an end-of-line or end-of-block character is recognized; available output is only sent after the end-of-block character. When input ends, the TIP positions the cursor to the lefthand margin of the next line on the display and unlocks the keyboard.

When normalized mode output is sent to a console, the TIP supports only one output mechanism, the display. Line folding and scrolling is assumed to be performed by the output mechanism when page width (see the PW command in section 5) is reached. When output ends, the TIP ensures that the cursor is positioned at the lefthand margin of the next line on the display. Page waiting is supported.

Transparent Mode

Transparent mode input and output are supported only for the display console. The TIP adds the line protocol frame control characters to output and removes them from input.

The TIP does not poll a terminal for console input after transparent mode input is received, until subsequent output is delivered to the console.

In transparent mode, the application program is responsible for handling all data interpretation, including any embedded IVT command or feature control characters. The application program must properly position the cursor at the end of input or output and must unlock the keyboard when necessary.

The TIP does not perform clear write or reset write operations at the beginning of transparent mode output. The device's transmission buffer is therefore not cleared and reset unless the proper control codes are part of the transparent output.

Terminal Code Sets and Parity

The MODE4 TIP supports terminals operating with a fixed parity, set to odd. Mode 4A terminals use the ASCII or External BCD code sets; mode 4C terminals use the ASCII code set.

The TIP converts terminal code sets to and from network ASCII codes where necessary. No ASCII control codes exist in normalized output to External BCD terminals; the TIP replaces all nongraphic characters with blanks and folds all lowercase characters to equivalent uppercase characters (see appendix A). The following ASCII codes effectively do not exist in normalized output to ASCII terminals:

- Control codes below the hexadecimal value 8 (these are converted to blanks)
- The control codes US and CR (hexadecimal 1F and 0D; these are converted to newline sequences)
- The control code LF (hexadecimal 1A; the TIP removes this code from the output)

In transparent mode, the TIP does not convert or delete characters in data sent to either External BCD or ASCII terminals.

The terminal code set used is initially established in the network configuration file. If the terminal supports the feature and alternate configurations exist in the network configuration file, the terminal user can change the code set prior to the automatic recognition process and initial connection to the network.

Normalized Mode Output

When the host sends ASCII character codes, the eighth upper bit (bit 7) can be anything and is ignored. If translation is needed, the value supplied by the host as bit 7 is ignored when the TIP translates the character. If translation is not needed, the bit is reset by the CLA as necessary. The CLA sets the bit to give the character odd parity.

Transparent Mode Output

The host can send anything for bit 7. Bit 7 is set correctly by the CLA for odd parity to ASCII or External BCD code terminals. The entire 8-bit byte sent from the host is never sent through the CLA to the terminal.

Normalized Mode Input

When the TIP sends the host ASCII character codes, the eighth bit is always set to 0.

Transparent Mode Input

The eighth bit is set to zero. The remaining 7 bits are sent to the host just as they came in on the communication line.

Initial Connection

The initial terminal class, protocol variant (mode 4A or mode 4C), cluster address, terminal address, device type, and code set can be explicitly set through the network configuration file or can be determined by the TIP through automatic recognition. The need for automatic recognition is established through the network configuration file.

Automatic recognition is performed when a line becomes active. The user must first press the transmission key (SEND or ETX). This action stores a read message in the cluster address for the terminal.

When the line becomes active, the TIP polls to find an active terminal. All cluster addresses are polled. If the first poll does not locate an active cluster address, a second poll occurs before the cluster is classified as failed hardware. Each poll is timed out for 1 to 1.5 seconds.

When an active terminal is found, the TIP polls that cluster address for a read message; the read message contains an escape code that indicates the code set used by the terminal. If the terminal uses External BCD, the automatic recognition process is complete; if the terminal uses ASCII, the terminal configuration is determined.

The TIP sends a configuration request to determine the devices present. If it receives error responses or no response after two attempts, the terminal is mode 4A. If the TIP receives a read response, the terminal is mode 4C, and the terminal addresses and device types of the individual devices are then determined.

The information received is passed to the Communications Supervisor in the host. The Communications Supervisor compares it against the information known for all possible terminals using the line, until a match is found. The Communications Supervisor then reports the configuration to the TIP. The mode 4C console device controlling a printer is the console with the same cluster address in the network configuration file that was the last console defined before the printer was defined.

Once the addressing information and the code set are known, the terminal connection to the NPU is complete. Subsequent connection of console and batch devices to a host and an application program depend on the network configuration; the procedure and possible commands required are described in section 4. Batch devices are always connected to the host to which the owning console is connected.

The security of a user's login information is ensured by the CCP's maintenance of a security control character (for each subTIP type).

The security character is site assigned (there is no default) during a CCP build. The character must be a 7-bit character within the code set of the terminal specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

Disconnection

The MODE4 TIP disconnects a terminal on a dial-up line when:

- The terminal uses automatic recognition information not recognized by the TIP.
- No devices are connected to a host for two minutes.
- The Communications Supervisor requests that the line or terminal be disabled.

When a console is disconnected from a host, the TIP disconnects any batch device for which that console is the owning device.

The MODE4 TIP does not disconnect terminals on a dedicated line. Input from disabled terminals or input on disabled hardwired lines is not requested.

Data Formatting in Normalized Mode

During input, the TIP detects the end of a physical line when an end-of-line or end-of-block indicator is entered.

During output to a console device, a physical line ends when the output ends or when it contains a carriage return or unit separator code.

Normalized Editing Modes

Only two of the editing modes described in section 2 are supported: complete editing and full-ASCII editing. Full-ASCII mode is similar to transparent mode, except the TIP continues its normal polling and protocol processing procedures.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input. For batch devices, polling ceases and input is stopped between transmission blocks whenever a temporary suspension of input must occur. For console devices, polling ceases.

Output from the host is delivered to the terminal during input regulation. Polling and input resumes as soon as regulation ends and data can again be stored by the NPU.

Error Recovery

The MODE4 TIP performs error recovery on input data by requesting retransmission of erroneous blocks from the terminal. The TIP detects the following errors:

An error indication from the terminal

No response from the terminal

An invalid address from the terminal

An invalid protocol response from the terminal

A character parity error

Data Carrier Detect signal dropped during input

An invalid MTI

An invalid or missing E code

A missing ETX or an ETX in an illegal sequence

A longitudinal parity check error

Lost write messages

Lost responses to write messages

The TIP determines whether the error occurred in the cluster controller or the device and attempts to correct the error by restarting the input/output sequence in which it occurred. This restart is called short-term recovery. Short-term recovery is attempted a fixed number of times (the default is 10); then the cluster or device is treated as failed hardware, and long-term recovery procedures begin.

If a parity error is detected during input, the TIP requests retransmission. The flag that becomes the pef bit in the host application block header is not used; data containing parity errors is discarded and replaced by correct data resent from the terminal.

Long term recovery differs for mode 4A and mode 4C terminals, and for cluster controller or device failures. For mode 4A terminals, cluster controller and device failure are identical, so separate device recovery is unnecessary; for mode 4C terminals, cluster controller and device failure are separate events.

Times indicated below are approximate. Only one attempt occurs during each cycle of servicing all the clusters and devices on the line, so that recovery attempts do not interfere with response time for other terminals using the line.

Cluster recovery on communication lines is attempted every few seconds (the default is 10 seconds for point-to-point communication lines and 50 seconds for multidrop lines). For mode 4A terminals, the TIP sends a clear write message to the console; for mode 4C terminals, the TIP sends a status request message to the cluster.

If the TIP gets a valid response, the cluster controller is operational. For mode 4C terminals, the TIP then performs recovery attempts on all devices within the cluster.

Device recovery attempts occur every few seconds (the default is 50 seconds). Device recovery is not attempted if the entire cluster must undergo recovery.

Device recovery consists of the TIP sending a clear write message to each mode 4C console or an empty write message to each mode 4C printer. If the TIP receives a valid response, the device is operational.

The toggle bit (bit 4 of the terminal address) is used during recovery of lost write messages or lost responses to write messages. The toggle bit value returned from a mode 4 terminal differs according to the terminal class.

Terminals in classes 10, 11, 13, and 15 return the toggle state of the last correctly received write message. These terminals do not distinguish the type of message to which the terminal is responding when they set or clear the toggle bit.

Terminals in class 12 return the toggle state of the last received message. These terminals return the toggle state used in poll messages, status requests, and so forth.

To compensate for this difference, the TIP initializes the toggle state of the terminal by sending a clear write message. This guarantees delivery of the first output transmission block.

When the TIP polls for a lost terminal response to a write message, the TIP sets the toggle state opposite to that of the last correctly received write message. If the toggle bit in the response is the same as in the polling message, the write message is sent again. This procedure guarantees that all write messages are correctly received by the terminal. No blocks are duplicated for terminals except 711s, because:

Blocks are only sent once to terminals in classes 10, 12, 13, and 15.

The Tektronix 4014 discards a write message with a toggle state that is the same as the previous write message.

Duplication of output during correction of a lost write message condition cannot be avoided at 711 devices. The 711 device does not maintain the correct toggle state and does not discard duplicate blocks.

HASP TIP

The HASP TIP supports bisynchronous terminals using the HASP multileaving protocol connected through dedicated or dial-up synchronous lines. Such terminals contain devices that exchange IVT data with a host application program in an interactive mode, and devices that exchange PRU data with a host application program in a batch mode.

The TIP supports one terminal per line but can support up to 22 devices per terminal. The TIP expects each terminal to have one interactive device, a console. The console must have an input and an output mechanism.

The terms terminal and device have varying definitions within HASP documentation. This subsection uses these definitions:

A terminal is equivalent to a workstation and is a set of devices with a single point of access to a communication line.

A device is a separately accessible unit of input or output equipment, such as a console, a card reader, a card punch, a plotter, or a line printer. Within the network, a device is separately addressable. For HASP terminals, all devices except the console have separate stream numbers. Each device is addressed by device type and stream number.

Stream numbers are described in section 3.

A CDC-defined HASP terminal has one console. It can also have up to seven card readers, up to seven line printers, and up to seven card punches or plotters.

The following CDC-defined terminal classes are supported:

Terminal Class	Archetype Terminal
9	HASP terminal supporting preprint format control
14	HASP terminal supporting only postprint format control

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the HASP TIP.

Each CDC-defined terminal class has a default set of terminal characteristics as shown in section 5. These terminal characteristics can be changed from their defaults initially for each communication line through the network configuration file or subsequently by the terminal user and/or the application program through the use of commands.

The valid values and default parameters for commands in each of these terminal classes are shown in section 5. When a valid value is entered from the console, the TIP responds with the message:

cd ACCEPTED..

where cd is the command mnemonic. When an invalid value is entered, the TIP responds with the message:

cd reason

The command represented by cd is ignored. The field identified as reason contains a message explaining the reason the command was ignored. Appendix B describes these messages.

The HASP TIP supports only the following IVT interface commands:

User break 1 and user break 2 (see the B1 and B2 commands in section 5)

Cancel input (see the CN command in section 5)

Change network control character (see the CT command in section 5)

Display terminal characteristics (see the CH command in section 5)

Lockout unsolicited console messages (see the LK command in section 5)

Change console page width (see the PW command in section 5)

Sending a message to the network operator (see the MS command in section 5)

Host node selection (see the HN and HS commands in section 4)

Host availability display (see the HD command in section 4)

Protocol Assumptions

This subsection describes those aspects of TIP operation that depend upon hardware use or are implementor-interpreted portions of the HASP communication protocol. These aspects include:

Stream identification

Filler use to maintain communication when data is not available for transmission

Terminal transparent transmissions

Line access

Operating mode

Stream Identification

The HASP TIP assumes that a network-assigned stream number between 1 and 7 corresponds to bits in the function control sequence (FCS). If the bits in the FCS are numbered from left to right, the following correspondences exist:

Card reader streams 1 through 4 are bits 4 through 7; card reader streams 5 through 7 are bits C through E.

Line printer streams 1 through 4 are bits 4 through 7; line printer streams 5 through 7 are bits C through E.

Card punch or plotter streams 7 through 5 are bits 5 through 7; card punch or plotter streams 4 through 1 are bits C through F.

The stream number algorithm allows separate flow control when more than eight printers, punches, and plotters are configured.

Filler Use

The TIP uses an ACK0 acknowledgment block as an idle block filler to transmit when no data is available to send downline. This idle block is transmitted at least every 2 seconds to prevent timeout on the communication line.

Terminal Transparent Transmissions

Two forms of terminal input and output are supported:

Terminal nontransparent transmissions

Terminal transparent transmissions

In either form, the TIP recognizes two modes of data:

Nontransparent network data

Transparent network data

The HASP TIP accepts terminal transparent transmissions beginning with the sequence DLE and STX. When such a transmission is received, the next output sent to the terminal occurs as a terminal transparent transmission. Nontransparent terminal output transmission resumes after the next nontransparent terminal input transmission.

Point-to-Point and Multidrop Line Access

The HASP TIP only supports nonaddressable HASP workstations on point-to-point communication lines.

Operating Mode

The HASP TIP is insensitive to line speeds.

The logical mode of operation of the TIP is half-duplex for each device. The TIP attempts to deliver console output whenever such data is available, unless an automatic input mode block has been delivered to the terminal.

The end of an input transmission block is signified by an ETB character code. This end-of-block indicator is part of the HASP protocol and cannot be changed.

Supported Input and Output Mechanisms

The TIP supports several input and output devices and mechanisms. Console devices have a keyboard as an input mechanism and a display as an output mechanism.

CDC-defined HASP batch devices are serviced concurrently with the console. All devices can be active at a given time, but only the input or output mechanism of the console device can be active at a given time. Input and output are therefore concurrent operations for batch devices and sequential operations for the console.

Card Reader Operation

Card reader operations are governed by special command blocks exchanged with the host. These commands originate within the PIP routine or are coverted by PIP to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Start input

Resume input

Abort input

The TIP sends command blocks to the host to:

Report when input stops

Report the end of the input stream

Report the number of cards read

Report the end of a file

A HASP card reader becomes ready for input when the host sends the TIP a start input command block. Once card input begins, it continues until:

An end-of-file (EOF) block is received from the card reader.

The host aborts input on the card reader stream.

The host disconnects or disables the terminal.

The workstation or communication line fails.

Card input begins with the first card read. A card input file begins when the beginning-of-information is detected. The TIP expects a nonblank data card as the beginning-of-information initially or following an end-of-information card. The TIP discards blank cards, end-of-record cards, or end-of-information cards read when it is expecting the beginning-of-information.

When the beginning-of-information is located, the TIP converts subsequent data into a NOS operating system job file in 6-bit display code. Conversion from 026 or 029 punch patterns is performed.

The initial punch pattern translation for each job file is determined by the network administrator through the network configuration file. The terminal user can change the translation for subsequent portions of the job file by punching a 26 or a 29 in columns 79 and 80 of any job card or of any end-of-record card that is not in a transparent portion of the file.

The TIP converts nontransparent card input data from EBCDIC to 6-bit display code. Compressed characters are uncompressed.

Trailing blanks are discarded unless the last nonblank character is a colon; one blank is preserved or inserted when the last character is a colon or the card contains an odd number of significant characters. For hosts using a 64-character set, this convention produces an 82-character record when the character in column 80 is a colon.

Every card image becomes a zero-byte terminated record. These records become a network block equivalent to one, two, or three physical record units (PRUs); a block equivalent to one PRU contains 640 6-bit characters or equivalent zero-byte terminator bytes, left-justified in an 8-bit character byte. A record spans two network blocks when necessary. In the host, NAM removes the two low-order bits and packs the characters into actual PRUs.

Unless the TIP is reading transparent card data, it recognizes /*EOR cards as end-of-record indicators for NOS system logical records. It converts any numbers found in columns 6 and 7 to octal record level numbers. Level number 17 is recognized as an end-of-file indicator within a multiple-file job. When necessary, short or zero-length PRUs are created.

Unless the TIP is reading transparent card data, it recognizes /*EOI cards as end-of-information indicators for NOS files. Once transparent input begins, the TIP recognizes only an end-of-file (EOF) block as end-of-information for the file.

Transparent card input (binary card data) is supported. The TIP does not translate subsequent portions of the job file when the user punches the characters TR in columns 79 and 80 of any end-of-record card. An end-of-record card containing such punches becomes the last recognized end-of-record card in the job file.

The TIP does not convert transparent input data to 6-bit display code. Transparent input characters are uncompressed and stored as 8-bit codes within 8-bit bytes. No record terminators are inserted. A record is continued in a subsequent network block when necessary.

These unterminated records become a network block equivalent to one, two, or three physical record units (PRUs). A block equivalent to one PRU contains 320 8-bit characters.

In the host, NAM packs the characters into actual PRUs. NAM right-justifies the 8-bit bytes received within 12-bit bytes for storage, zero-filling bits 8 through 11. Transparent card input becomes a variable number of variable length unterminated records containing five characters per word in each physical record unit of 64 60-bit words.

Printer Operation

Line printer operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report when a file limit is exceeded

Report the number of lines printed

Report a printer message (PM) in the output

Report the end of a file

The TIP begins output to a printer when it has output queued for the device. Data received from the host is packed as line images in PRU-sized network blocks (the host PIP routine strips zero-byte terminators and replaces them with hexadecimal FF code bytes). The TIP discards trailing blanks on a print line. End-of-record or end-of-file indicators in the data terminate an output transmission block.

Six-bit display code is converted to EBCDIC. ASCII print files are converted from 7-bit codes right-justified in 8-bit bytes to the EBCDIC codes supported for normalized mode data (see Terminal Code Sets and Parity). This conversion includes stripping the unneeded bit 7. The host PIP routine removes the unneeded bits 8 through 11 of the stored 12-bit byte before transmission downline.

Printer busy and not-ready conditions are not reported to the host. Printing continues when the condition clears.

Once started, the TIP continues output to printers unless stopped or interrupted by a command block from the host or a printer message (PM carriage control sequence in the print file). Any printer that is interrupted or stopped can be restarted by a command block from the host.

The TIP supports both preprint and postprint printers. Different format effector substitutions occur, depending on the terminal class of the workstation. The TIP uses the subrecord control byte (SRCB) content appropriate to the terminal class.

Some HASP printers cannot suppress carriage control. These printers sometimes generate extra spaces in output and do not overprint lines.

Print Message Processing

When the print file contains a line beginning with the characters PM, the TIP stops the printer, notifies the application program through a command block, and sends the data from the print file line to the application program.

Card Punch Operation

Card punch operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report when a file limit is exceeded

Report the number of cards punched

Report the end of a file

The TIP begins output to a punch when it has output queued for the device. Data received from the host is packed as card images in PRU-sized network blocks (the host PIP routine strips zero byte terminators and replaces them with hexadecimal FF code bytes).

End-of-record or end-of-file indicators in the data terminate an output transmission block. The TIP generates 7/8/9 multiple punch cards to mark end-of-record indicators, and /*EOI cards to mark end-of-file indicators.

Six-bit display code is converted to EBCDIC. Punch pattern translation is selected by a command block preceding the data. If no such comand block is received, 029 punch patterns are assumed.

Punch busy and not-ready conditions are not reported to the host. Punching continues when the condition clears.

Once started, the TIP continues output to punches unless stopped or interrupted by a command block from the host. Any punch that is interrupted or stopped can be restarted by a command block from the host.

Plotter Operation

Plotter operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report when a file limit is exceeded

Report the number of plotter records delivered

Report the end of a file

The TIP begins output to a plotter when it has output queued for the device. Data is received from the host in PRU-sized network blocks. End-of-record or end-of-file indicators in the data terminate an output transmission block.

Plotter data is always in network transparent mode. Plotter data consists of 6-bit or 8-bit binary codes and is therefore not converted to EBCDIC by CCP. PIP right-justifies 6-bit codes within the 8-bit character byte, with zero fill. The 8-bit codes in the host are stored as five characters per word, right-justified within 12-bit bytes with zero fill; PIP strips the unneeded bits 8 through 11 before transmission to CCP.

The data is formatted into HASP records by the TIP; normal data compression occurs. Each record ends when it contains the number of characters established as the page width of the device, or when the end of the network block is reached and the block is an end-of-record or end-of-information block. The initial page width is established by the network administrator in the network configuration file.

Plotter busy and not-ready conditions are not reported to the host. Plotting continues when the condition clears.

Once started, the TIP continues output to plotters unless stopped or interrupted by a command block from the host. Any plotter that is interrupted or stopped can be restarted by a command block from the host.

Normalized Mode Console Operation

The TIP supports only normalized mode input from a console, using only one input mode and mechanism: the keyboard in line mode. Input is forwarded after each console record is received (after each end-of-block indicator is recognized); available output is only sent after the end-of-block character.

The TIP supports only normalized mode output to a console, using only one output mechanism: the display. Line folding and scrolling is assumed to be performed by the output mechanism when page width (see PW command in section 5) is reached. When output ends, the TIP ensures that the cursor is positioned at the lefthand margin of the next line on the display.

Transparent Mode

Network transparent mode input and output are only supported for batch devices. Network transparent input is allowed only from card readers. Network transparent output is allowed only to plotters, and is required for them.

The TIP does add the line protocol frame control characters to network transparent output and removes them from network transparent input. In network transparent mode, the application program is responsible for handling all data interpretation, including any embedded control characters.

Terminal Code Sets and Parity

The HASP TIP supports terminals operating without parity. HASP terminals use the IBM EBCDIC code set.

The TIP converts terminal code sets to and from network ASCII codes where necessary. During normalized output to HASP terminals, the TIP replaces ASCII characters that have no EBCDIC equivalents with blanks. During normalized input from HASP terminals, the TIP replaces EBCDIC characters that have no ASCII equivalents with blanks (see appendix A).

The card reader network input mode is initially nontransparent. The terminal user can change the mode to network transparent by using the characters TR on end-of-record cards in an input deck. This process is described earlier in this subsection under Card Reader Operation.

In network transparent mode, the TIP does not convert or delete characters in data.

The plotter output mode is fixed as network transparent. Plotter data can be either 6-bit instruction code bytes or 8-bit instruction code bytes. The TIP does not alter the byte content.

Normalized Mode and Network Nontransparent Mode Output

When the host sends ASCII character codes, the eighth upper bit (bit 7) can be anything and is ignored when the TIP translates the character. The bit is set as needed for the corresponding EBCDIC character.

Network Transparent Mode Output

The host can send anything for bit 7. Bit 7 is not changed by the CLA or the TIP. The entire 8-bit byte sent from the host is sent through the CLA to the terminal.

Normalized Mode and Network Nontransparent Mode Input

When the TIP sends the host ASCII character codes, the eighth bit is always set to 0.

Network Transparent Mode Input

The eighth bit (bit 7) of the data is not changed. All eight bits are sent to the host just as they came in on the communication line.

Initial Connection

The initial terminal class, stream numbers, configuration ordinal, and device types can be explicitly set through the network configuration file or can be determined by the TIP through automatic recognition. The need for automatic recognition is established through the network configuration file.

Automatic recognition is performed when a line becomes active. This process occurs as part of the initial communication sequence between the TIP and the terminal, as follows:

The terminal sends an enquiry block (ENQ).

The TIP responds with an acknowledgment block (ACK0).

The terminal sends a signon block on the console stream. The signon block contains either a /*SIGNON card image or a /*CONFIG card image.

If automatic recognition is required and the TIP receives a /*SIGNON card or an invalid /*CONFIG card, it discards the card and sends the diagnostic message

image <BAD CONFIG>

to the terminal, where image contains the portion of the card that cannot be processed.

The TIP waits for input of a valid /*CONFIG card from a card reader. The format of a valid /*CONFIG card is given in section 4.

When a valid /*CONFIG card is received, the information is passed to the Communications Supervisor in the host. The Communications Supervisor compares it against the information known for all possible terminals using the line, until a match is found. The Communications Supervisor then reports the configuration to the TIP.

Once the signon block or /*CONFIG card is successfully processed, the terminal connection to the NPU is complete. Subsequent connection of the console and batch devices to a host and an application program depend on the network configuration; the procedure and possible commands required are described in section 4. Batch devices are always connected to the host to which the console is connected.

The security of a user's login information is ensured by the CCP's maintenance of a security control character (for each subTIP type).

The security character is site assigned (there is no default) during a CCP build. The character must be a 7-bit character within the code set of the terminal specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

If automatic recognition is not required for the communication line, the signon block is ignored and any /*CONFIG card image received by the TIP is sent to the host as data. If the card image is received on the console stream, it is sent upline on the console connection. If the card image is received on a card reader stream, it is sent upline as the first card of a job file.

Disconnection

The HASP TIP does not recognize a /*SIGNOFF card as a method of disconnecting a terminal. A /*SIGNOFF card read as the only card in a job file is sent to the host as data.

The HASP TIP disconnects a terminal on a dial-up or a dedicated line when:

The terminal is not connected to a host for two minutes.

The Communications Supervisor requests that the line or terminal be disabled.

A line or terminal failure is detected.

When a console is disconnected from a host, the TIP disconnects all batch devices in the same terminal.

Data Formatting in Normalized Mode

During console input, the TIP detects the end of a physical line when:

An end-of-line indicator is detected; this end-of-line indicator is the end-of-record for the console stream.

During output to a console display device, a physical line ends when the output ends or when it contains a carriage return or unit separator code.

Normalized Editing Modes

Only one of the editing modes described in section 2 is supported: complete editing.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input.

The HASP protocol provides two levels of input regulation:

A global wait bit

Individual FCS stream bits

For devices assigned to a regulation level of priority 3 (normally batch devices), one or more of the bits in the FCS is set to zero, and input is stopped between transmission blocks whenever a temporary suspension of input must occur. For devices assigned to a regulation level of priority 2 (normally console devices), only the wait bit is set; this action suspends all device input between transmission blocks.

Output from the host is delivered to the terminal during regulation. The FCS or the wait bits are changed, and input resumes as soon as regulation ends and data can again be stored by the NPU.

Output Regulation

The terminal can also suspend output to a single device by using the FCS bits or to all devices by using the wait bit. The HASP TIP recognizes either use. When an FCS bit is used, the TIP presumes the wait bit is not used. When the wait bit is used, the TIP presumes the FCS bits are unchanged.

Error Recovery

The HASP TIP performs error recovery on input data by requesting retransmission of erroneous blocks from the terminal. The TIP detects the following errors:

Cyclic redundancy count errors

Illegal transmission block formats

Unknown responses

Timeout over the communication line

Block control byte (BCB) errors, such as a break in the sequence of transmitted blocks

When the TIP receives a negative acknowledgment block (NAK) that indicates an error in reception of downline data, the TIP attempts to retransmit the block 62 times. If the sixty-second attempt to transmit fails, the TIP classifies the communication line and terminal inoperative.

The TIP requests retransmission of a bad upline block 63 times. If the sixty-third attempt to receive fails, the TIP classifies the communication line and terminal inoperative.

Because EBCDIC does not contain a parity bit, the TIP cannot detect a parity error. It does not store erroneous code in the network block or set a flag for such data. The flag that becomes the pef bit in the host application block header is not used because data containing parity errors is discarded and replaced by correct data resent from the terminal.

BSC TIP

The BSC TIP supports IBM-2780- or IBM-3780-compatible bisynchronous terminals connected through dedicated or dial-up synchronous lines. Such terminals can contain devices that generate IVT data to exchange with a host application program in an interactive mode, but normally contain devices that exchange PRU data with a host application program in a batch mode.

The TIP supports more than one terminal device per line. A 2780 or 3780 terminal always contains one card reader and one line printer; one card punch can also be supported.

The host software expects each terminal to have at least one interactive device, serviced through the interactive virtual terminal interface. The interactive device must have an input and an output mechanism. Because 2780 and 3780 terminals normally do not have consoles, the TIP simulates the existence of a console by using a special data mode for batch devices. The console simulation method is the reason each terminal must have one card reader and one line printer. The card reader is serviced as an interactive input mechanism and the line printer as an interactive output mechanism.

The terms terminal, terminal address, and device have varying or no definitions within 2780 or 3780 documentation. This subsection uses these definitions:

A terminal is a set of devices with a single point of access to a communication line.

A terminal address is the control character code used to identify a card punch transmission block.

A device is a separately accessible unit of input or output equipment, such as a console, a card reader, or a line printer. Within the network, a device is separately addressable, regardless of whether it has a separate hardware address within the terminal.

Terminal addresses are described in section 4, where the /*CONFIG statement is defined.

The following CDC-defined terminal classes are supported:

Terminal Class	Archetype Terminal
16	IBM 2780 remote job entry terminal
17	IBM 3780 remote job entry terminal

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the BSC TIP.

Each CDC-defined terminal class has a default set of terminal characteristics as shown in section 5. These terminal characteristics can be changed from their defaults initially for each communication

line through the network configuration file or subsequently by the terminal user and/or the application program through the use of commands.

The valid values and default parameters for commands in each of these terminal classes are shown in section 5. When a valid value is entered from the console, the TIP responds with the message:

cd ACCEPTED..

where cd is the command mnemonic. When an invalid value is entered, the TIP responds with the message:

cd reason

The command represented by cd is ignored. The field identified as reason contains a message explaining the reason the command was ignored. Appendix B describes these messages.

The BSC TIP supports only the following IVT interface commands and features:

Changing the network control character (see the CT command in section 5)

Changing the page width of the console (see the PW command in section 5)

Displaying terminal characteristics (see the CH command in section 5)

Sending a message to the network operator (see the MS command in section 5)

Host node selection (see the HN and HS commands in section 4)

Host availability display (see the HD command in section 4)

Protocol Assumptions

This subsection describes those aspects of TIP operation that depend upon hardware use or are implementor-interpreted portions of 2780 or 3780 communication subprotocols. These aspects include:

Terminal feature support

Terminal transparent transmissions

Size and number of records per transmission block from or to a terminal

Resolving line contention

Line access

Operating mode

Terminal Features

The following 2780 and 3780 features are supported:

Redundancy checking

Extended ENQ retransmissions

Automatic line turnaround

Automatic disconnection

Blank compression and expansion (2780 support is separately described in the next subsection)

Postprint carriage control

Transparent transmission format, separate from network transparent mode

The following 2780 terminal features are supported:

Multiple record transmission blocks; from 2 to 7 records per block can be configured by the site administrator

80-, 120- and 144-character print lines

Batch job file termination when column 80 of the last card read contains an ETX code punch

The following 3780 terminal features are supported:

Multiple record transmission blocks in transparent transmission format

Punch component selection code

The following features are not supported:

Character code set selection and USASCII (ASCII) or 6-bit transcode code use

Preprint carriage control

Horizontal tab control

Automatic answer terminals on dialup lines

Audible alarm

Reverse interrupt

Conversational mode

2780 Models 3 and 4

2780 Blank Compression

During input, blank compression is supported as follows:

Two or more consecutive blanks are received as a BYP code (24 hexadecimal) followed by a biased count of the number of compressed blanks; a bias of hexadecimal 80 is used, and the unbiased count cannot exceed 31 (1F hexadecimal). For example, 11 blanks becomes 248B hexadecimal.

When more than 31 (1F hexadecimal) consecutive blanks exist, multiple BYP and count sequences are used.

A 12-2-9 punch in column 1 (an STX code) inhibits blank compression for the rest of a card reader record.

Compression is allowed in terminal and network transparent modes.

During output, blank compression is supported as follows:

Terminal transparent mode compressed blanks expand to 80-character punch records or printer records of the defined batch device page width.

A 12-2-9 punch in column 1 inhibits blank decompression for the rest of a punch record. That record cannot contain a BYP code and cannot contain an STX code as its first data character.

Terminal Transparent Transmissions

Two forms of terminal input and output are supported:

Terminal nontransparent transmissions

Terminal transparent transmissions

In either form, the TIP recognizes two modes of data:

Nontransparent network data

Transparent network data

When a terminal begins communicating with the network, it uses nontransparent network data.

The BSC TIP accepts terminal transparent transmissions beginning with the sequence DLE and STX. When such a transmission is received, the next output sent to the terminal occurs as a terminal transparent transmission. Nontransparent terminal output transmission resumes after the next nontransparent terminal input transmission.

Record Blocking

A 2780 terminal using nontransparent transmissions sends or receives two (a default value, changeable by the network adminstrator through the network configuration file) 80-character fixed-length records or up to 7 variable-length records per block. A 3780 terminal using nontransparent transmissions sends or receives a maximum number of characters per transmission block, regardless of the number of records in the block. The number of 3780 characters used by the TIP during output also is initially established by the network administrator in the network configuration file.

In transparent input transmissions, records always contain 80 characters. In nontransparent input transmissions, records contain from 1 through 80 characters.

Line Contention

Contention for line use is resolved in favor of input data. Batch output can be interrupted to accept card reader batch data or console data. Interrupted output is restarted when the end-of-transmission is detected in input. Card reader input is not interruptable.

Point-to-Point and Multidrop Line Access

Nonaddressable 2780 and 3780 terminals are supported on point-to-point communication lines only. Terminal identifiers in line bids are accepted but not checked.

Operating Mode

The BSC TIP is insensitive to line speeds.

The logical mode of operation of the TIP is half-duplex for each device, with console input given priority over output. This means that console output is not started when input is active.

The TIP attempts to deliver console output whenever such data is available. Automatic input mode blocks are ignored by the BSC TIP.

Supported Input and Output Mechanisms

The TIP supports three batch input and output devices. Real console devices must emulate card reader input and accept line printer output if used.

Sequential Operation

For BSC terminals, only one input or one output mechanism can be active at any given time. Input and output are therefore sequential operations.

The TIP suspends console output when either the card reader becomes active or data is available for the line printer.

Batch input and output alternate unless one of the following events occurs:

Output arrives for the console from the host; this output is printed as soon as the current batch print file ends.

The card reader or printer is stopped (or suspended).

Console input begins (the first two characters of a card are /* and the card reader hopper was previously empty).

Card Reader Operation

Card reader operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Start input

Resume input

Abort input

The TIP sends command blocks to the host to:

Report when input stops

Report the end of the input stream

Report the number of cards read

Report the end of a file

A 2780 or 3780 card reader becomes ready for batch input when the host sends the TIP a start input command block. Card input begins with the first card read.

A card input file begins when the beginning-of-information is detected. The TIP expects a nonblank data card as the beginning-of-information initially or following an end-of-information card. The TIP discards blank cards, end-of-record cards, or end-of-information cards read when it is expecting the beginning of information.

When the beginning-of-information is located, the TIP converts subsequent data into a NOS operating system job file in 6-bit display code. Conversion from 026 or 029 punch patterns is performed.

The initial punch pattern translation for each job file is determined by the network administrator through the network configuration file. The terminal user can change the translation for subsequent portions of the job file by punching a 26 or a 29 in columns 79 and 80 of any job card or any end-of-record card that is not in a transparent portion of the file.

The TIP converts network nontransparent card input data from EBCDIC to 6-bit display code. Compressed blanks and zeros are uncompressed.

Trailing blanks are discarded unless the last nonblank character is a colon; one blank is preserved or inserted when the last character is a colon or the card contains an odd number of significant characters. For hosts using a 64-character set, this convention produces an 82-character record when the character in column 80 is a colon.

Every card image becomes a zero-byte terminated record. These records become a network block equivalent to one, two, or three physical record units (PRUs); a block equivalent to one PRU contains 640 6-bit characters or equivalent zero-byte terminator bytes, left-justified in an 8-bit character byte. A record is continued in a subsequent network block when necessary. In the host, NAM removes the two low-order bits and packs the characters into actual PRUs.

Unless the TIP is reading network transparent card data or the terminal is sending terminal transparent transmissions, the TIP recognizes /*EOR cards as end-of-record indicators for NOS system logical records. It converts any numbers found in columns 6 and 7 to octal record level numbers. Record level 17 is recognized as an end-of-file indicator. When necessary, short or zero-length PRUs are created.

Unless the TIP is reading network transparent card data or the terminal is sending terminal transparent transmissions, the TIP recognizes /*EOI cards as end-of-information indicators for NOS files. If the terminal is sending transparent transmissions, the TIP also recognizes an ETX code punched in column 80 of the last card read (or in column 1 of an otherwise blank card) as an end-of-information indicator for a NOS file. Once transparent data input begins, the TIP recognizes only an end-of-transmission code (EOT) as an end-of-information for the file.

Network transparent card input (binary card data) is supported. The initial translation mode is always network nontransparent. The TIP does not translate subsequent portions of the job file when the user punches the characters TR in columns 79 and 80 of any end-of-record card. An end-of-record card containing such punches becomes the last recognized end-of-record card in the job file.

The TIP does not convert network transparent input data to 6-bit display code. Network transparent input characters are stored as 8-bit codes within 8-bit bytes. No record terminators are inserted. Compressed blanks are not expanded. A record spans two network blocks when necessary.

These unterminated records become a network block equivalent to one, two, or three physical record units (PRUs). A block equivalent to one PRU contains 320 8-bit characters.

In the host, NAM packs the characters into actual PRUs. NAM right-justifies the 8-bit bytes received within 12-bit bytes for storage, zero-filling bits 8 through 11. Network transparent card input becomes a variable number of variable length unterminated records containing five characters per word in each physical record unit of 64 60-bit words.

Printer Operation

Line printer operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report when a file limit is exceeded

Report the number of lines printed

Report a printer message (PM) in the output

Report the end of a file

The TIP begins output to a printer when it has output queued for the device and no input is in process. Data received from the host is packed as line images in PRU-sized network blocks (NAM strips zero-byte terminators and replaces them with hexadecimal FF code bytes). The TIP discards trailing blanks on a print line. End-of-record or end-of-file indicators in the data terminate an output transmission block. The end of a file is also marked by sending an ETX block to the terminal.

Six-bit display code is converted to EBCDIC. ASCII print files are converted from 7-bit codes right-justified in 8-bit bytes to the EBCDIC codes supported for normalized mode data (see Terminal Code Sets and Parity). This conversion includes stripping the unneeded bit 7. NAM strips the unneeded bits 8 through 11 from the 12-bit byte used for storage before sending the data downline.

Printer busy and not-ready conditions are not reported to the host. Printing continues when the condition clears.

Once started, the TIP continues output to printers unless stopped or interrupted by a command block from the host or a printer message (PM carriage control sequence in the print file). Any printer that is interrupted or stopped can be restarted by a command block from the host.

Print Message Processing

When the print file contains a line beginning with the characters PM, the TIP stops the printer, notifies the application program through a command block, and sends the data from the print file line to the application program.

Card Punch Operation

Card punch operations are governed by special command blocks exchanged with the host. These commands originate within NAM or are converted by NAM to or from supervisory messages exchanged with an application program. The host sends commands to the TIP to:

Terminate output

Mark the point of output termination

Restart output

The TIP sends command blocks to the host to:

Report when a file limit is exceeded

Report the number of cards punched

Report the end of a file

The TIP begins output to a punch when it has output queued for the device. Data received from the host is packed as card images in PRU-sized network blocks (NAM strips zero-byte terminators and replaces them with hexadecimal FF code bytes).

End-of-record or end-of-file indicators in the data terminate an output transmission block. The TIP generates /*EOR cards to mark end-of-record indicators, and /*EOI cards to mark end-of-file indicators. The end of a file is also marked by sending an ETX block to the terminal.

Six-bit display code is converted to EBCDIC. Punch pattern translation is selected through a command block that precedes the data. If no command block is received, 029 punch patterns are assumed.

The TIP prefixes the first card image in each punch file with the terminal address code associated with the punch. For 3780 terminals, this code is initially established by the network administrator in the network configuration file. The code used for a 3780 can also be established by the terminal user during the automatic recognition process, described later in this subsection.

Punch busy and not-ready conditions are not reported to the host. Punching continues when the condition clears.

Once started, the TIP continues output to punches unless stopped or interrupted by a command block from the host. Any punch that is interrupted or stopped can be restarted by a command block from the host.

Normalized Mode Console Operations

The TIP supports only normalized mode input from a console, using only one input mode and mechanism: the card reader, emulating a keyboard in line mode. Console input begins with the characters

/*

in character positions 1 and 2 of a card image. Input is forwarded after the card ends and the end-of-transmission character is recognized; available output is only sent after the end-of-transmission character.

Console output is sent only after output of a batch file is completed. The first line of any console output is prefixed by the message described in section 2.

The TIP supports only normalized mode input from a console, using only one output mechanism: the line printer, emulating a display. Line folding and paging is performed using carriage control characters when page width (see the PW command in section 5) is reached.

Network Transparent Mode

Network transparent mode input is only supported for the card reader. The TIP does remove the line protocol frame control characters from input.

In network transparent mode, the application program is responsible for handling all data interpretation, including any embedded control characters.

Terminal Code Sets and Parity

The BSC TIP supports terminals operating without parity. IBM 2780 and 3780 terminals use the EBCDIC code set.

The TIP converts terminal code sets to and from network ASCII codes where necessary. During normalized output to BSC terminals, the TIP replaces ASCII characters that have no EBCDIC equivalents with blanks. During normalized input from BSC terminals, the TIP replaces EBCDIC characters that have no ASCII equivalents with blanks (see appendix A).

In network transparent mode, the TIP does not convert or delete characters in data.

The card reader input mode is initially network nontransparent. The terminal user can change the mode to network transparent by using the characters TR on certain cards in an input deck. This process is described earlier in this subsection under Card Reader Operation.

Normalized Mode and Network Nontransparent Mode Output

When the host sends ASCII character codes, the eighth bit (bit 7) can be anything and is ignored

when the TIP translates the character. The bit is set as needed for the corresponding EBCDIC character.

Network Transparent Mode Output

The host can send anything for bit 7. Bit 7 is not changed by the CLA or the TIP. The entire 8-bit byte sent from the host is sent through the CLA to the terminal.

Normalized Mode and Network Nontransparent Mode Input

When the TIP sends the host ASCII character codes, the eighth bit is always set to 0.

Network Transparent Mode Input

The eighth bit (bit 7) is not changed. All eight bits are sent to the host just as they came in on the communication line.

Initial Connection

The initial terminal class, terminal address code for the punch, configuration ordinal, and device types can be explicitly set through the network configuration file or can be determined by the TIP through automatic recognition. The need for automatic recognition is established through the network configuration file.

Automatic recognition is performed when a line becomes active. This process occurs as part of the initial communication sequence between the TIP and the terminal, as follows:

The terminal sends an enquiry block (ENQ).

The TIP responds with an acknowledgment block (ACK0).

The terminal sends a /*CONFIG card image.

If automatic recognition is required and the TIP receives an invalid /*CONFIG card, it discards the card and sends the diagnostic message

image <BAD CONFIG>

to the terminal, where image contains the portion of the card that cannot be processed. The TIP waits for input of a valid /*CONFIG card from a card reader. The format of a valid /*CONFIG card is given in section 4.

When a valid /*CONFIG card is received, the information is passed to the Communications Supervisor in the host. The Communications Supervisor compares it against the information known for all possible terminals using the line, until a match is found. The Communications Supervisor then reports the configuration to the TIP.

Once the /*CONFIG card is successfully processed, the terminal connection to the NPU is complete. Until the processing is completed, the TIP discards any input received from the terminal. Subsequent connection of the console and batch devices to a host and an application program depends on the network configuration; the procedure and possible commands required are described in section 4. Batch devices are always connected to an application program in the host to which the console is connected.

The security of a user's login information is ensured by the CCP's maintenance of a security control character (for each subTIP type).

The security character is site assigned (there is no default) during a CCP build. The character must be a 7-bit character within the code set of the terminal specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

If automatic recognition is not required for the communication line, any /*CONFIG card image received by the TIP is sent to the host as data. Because the card image is received on the console stream, it is sent upline on the console connection.

Disconnection

The BSC TIP disconnects a terminal on a dial-up line when:

The terminal uses automatic recognition information not recognized by the TIP.

The terminal user fails to complete the automatic recognition sequence within the time allowed.

The terminal is not connected to a host for two minutes.

The Communications Supervisor requests that the line or terminal be disabled.

The BSC TIP does not disconnect a terminal on a dedicated line. Input from disabled terminals or input on disabled hardwired lines is discarded. Failure to complete automatic recognition on a dedicated line causes the TIP to restart the procedure.

Data Formatting in Normalized Mode

During input from the emulated console, the TIP detects the end of a physical line when a card image ends and the end-of-transmission code is received. During output to the emulated console device with a display mechanism, a physical line ends when the output ends, when the page width is reached, or when the output contains a carriage return or line feed code.

Normalized Editing Modes

Only one of the editing modes described in section 2 is supported: complete editing.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input. If input is in process when regulation begins, the TIP sends a wait/acknowledgment message (WACK) to the terminal; this action suspends the input between transmission blocks whenever a temporary suspension of input must occur. The data transmitted from the terminal is not discarded. If the terminal is idle when regulation begins, the TIP prevents subsequent input by sending an end-of-transmission (EOT) code to the terminal.

Output from the host is sent to the terminal during regulation. When regulation ends and data can again be stored by the NPU, the TIP sends an enquiryblock (ENQ) to resume input at the point where it was suspended.

Error Recovery

The BSC TIP performs error recovery on input data by requesting retransmission of erroneous blocks from the terminal. The TIP detects the following errors:

Cyclic redundancy count errors

Illegal transmission block formats

Unknown responses

Timeout over the communication line

When the TIP receives a negative acknowledgment block (NAK) that indicates an error in reception of downline data, the TIP attempts to retransmit the block three times. If the fourth attempt to transmit fails, the TIP classifies the communication line and terminal inoperative.

The TIP requests retransmission of a bad upline block three times. If the fourth attempt to receive fails, the TIP classifies the communication line and terminal inoperative.

Because EBCDIC does not contain a parity bit, the TIP cannot detect a parity error. The flag that becomes the pef bit in the host application block header is not used because data containing parity errors is discarded and replaced by correct data resent from the terminal.

3270 TIP

The 3270 TIP supports terminals that use the bisynchronous, 3270 polled protocol connected through dedicated or dial-up synchronous lines. Such terminals exchange IVT data with host application programs in an interactive mode, and they exchange PRU data with a host application program in a batch mode.

The TIP expects each terminal to have at least one interactive device. The interactive device must have an input and an output mechanism.

Cluster and device are terms having several definitions within 3270 publications:

A cluster is defined as a group of devices with a single cluster address and a controller. A 3270 cluster can have one or more consoles and zero or more printers.

A device is defined as a separately accessible unit of input or output, such as a console or a line printer. Within the network, a device is separately addressable, whether or not it has a separate hardware address within the terminal.

The following CDC-defined terminal class is supported:

Terminal Class	Archetype Terminal
18	IBM 3270

Allowed variants are:

3271 Control Unit, Models 1 and 2

3274 Control Unit, Model 1C

3276 Control Unit/Display Station, Models 1, 2, 3, and 4

3277 Display Station, Models 1 and 2

3278 Dsiplay Station, Models 1 and 2

Site-defined devices in terminal classes 28 through 31 can also be supported if the site modifies the released version of the 3270 TIP.

Each CDC-defined terminal class has a default set of terminal characteristics (as shown in section 5).

These terminal characteristics can be changed from their initial default values by using the network configuration file or, subsequently, by the terminal user and/or the application program using the command set.

The values and default parameters for the commands in each of these terminal classes are described in section 5. When a (valid) value is entered from the console, the TIP responds with the message:

cd ACCEPTED..

where cd is the command mnemonic. When an invalid value is entered, the TIP responds with the message:

cd reason

The command represented by cd is ignored. The field identified as reason contains a message explaining why the command was ignored. Appendix B describes these messages.

The 3270 TIP supports all of the IVT interface commands and features except the following:

Abort output blocks (the AB command)

Automatic recognition restart (the AR command)

Backspace processing (the BS command)

Break condition as user break or cancel input (the BR command)

Input and output regulation of or by devices, called device control (the IC and OC commands)

Idle character insertion at the end of physical lines (the CI and LI commands)

Echoplexing (the EP command)

Paper tape input or output (the IN command, PT and XP options, and the OP command PT option)

Parity selection (the PA command)

Special editing (the SE command)

Console printing mechanism selection (the OP command)

Cursor positioning (the EL and EB command CP option or the CP command)

Changing the end-of-block character code to values other than the default (the EL and EB commands)

Transparent mode timeout or character count delimiter, or changing the character code used as a delimiter (the DL or the XL command TO or C option, or the X and Y options)

Protocol Assumptions

This subsection describes TIP operations that depend either upon hardware use or are implementor-interpreted portions of the 3270 bisynchronous protocol. These aspects include:

Unsupported protocol features

Polling

Live access

Operating mode

Character set

Unsupported Protocol Features

The 3270 TIP does not support the following bisynchronous features:

TTD

Forward abort

Conversational mode

3275 dial-feature (contention protocol)

Polling

The 3270 TIP, under normal conditions, issues one general poll per second per cluster.

Any device having an input requirement can be internally selected by the 3270 cluster controller.

The cluster controller starts at a random device and inputs all device messages, sequentially, as long as ACKs are sent by the TIP and until all devices have been serviced. The first two characters of the first block of each input message identify the responding device.

The input process terminates when the cluster controller indicates that it has no further traffic.

An output device will be selected when output is both available and the device is ready to receive.

Line Access

The 3270 TIP controls activity on a 3270 terminal by polling for input and selecting for delivery of output.

The 3270 terminal is under control of the TIP in either of two modes:

control mode

text mode

In control mode, the terminal is monitoring the line for a valid poll/select sequence. When such a sequence is detected, the terminal enters the text mode.

In text mode, the terminal is either the master or the slave, with the TIP assuming the opposite role. When the entry into text mode is the result of a poll, the terminal is the master. When transmission is caused by a select, the TIP is the master.

In text mode, blocks of data are transferred from the master to the slave, one at a time, with a positive acknowledgment for each block being required before the next block is transferred. The master normally indicates the end of the transfer and sends an end-of-transmission sequence that initiates a return to control mode.

Operating Mode

The 3270 TIP is insensitive to line speeds. The logical mode of operation of the TIP is half-duplex for each device, with console input having priority over batch output. The TIP attempts to deliver console output whenever such data is available, unless a page wait condition is in effect.

The end of an input transmission block is an ETX or ETB character code. This end-of-block indicator is part of the bisynchronous protocol and it cannot be changed.

Supported Input and Output Mechanisms

The TIP supports several input and output devices and mechanisms. Console devices have a keyboard as an input mechanism and a display as an output mechanism.

The 3270 batch printers are serviced concurrently with the console.

Concurrent Operation

All devices on a cluster can be concurrently active, but only the input or the output mechanism for the console device can be active. Input and output are therefore concurrent operations for batch devices but sequential operations for the console.

Interactive operations are concurrent with batch operations.

Printer Operation

Line printer operations are governed by special command blocks exchanged with the host. These command blocks either originate within NAM or are converted by NAM to (or from) supervisory messages. The supervisory messages are exchanged with the application program. The host sends the commands to the TIP in order to:

Terminate output

Mark the point of output termination

Restart output

The TIP then sends the command blocks to the host in order to:

Report when a file limit is exceeded

Report when a printer message (PM) is received

Report the number of lines printed

Report the end of a file

The TIP initiates output to a printer when output is queued for the device. Data received from the host is packed as line images in PRU-sized network blocks (NAM strips off the zero byte terminators and replaces them with hexadecimal FF code bytes). The TIP discards trailing blanks on a print line.

Six-bit display code is converted to EBCDIC. ASCII print files are converted from 7-bit codes, right-justified in 8-bit bytes, to EBCDIC. This conversion includes the stripping out of the unneeded bit 7 (the host PIP routine strips the unneeded bits 8 through 11 prior to downline transmission).

Printer busy and not-ready conditions are not reported to the host, and printing continues when the condition clears.

Once started, the TIP continues output to printers unless stopped, or interrupted, by a command block from the host or for a printer message (for example, PM carriage control sequence in print file). Any printer that is interrupted, stopped, or not ready, can be restarted by a command from the host.

Print Message Processing

When the print file contains a line beginning with the characters PM, the TIP stops the printer, notifies the application program of this code through a command block, and sends the data from the print file line to the application program.

Normalized Mode Console Operations

When normalized mode input is in effect from a console, the TIP can support only one input mode and one mechanism (refer to the IN command in section 5): the keyboard is in line mode. Input is forwarded after an end-of-line character is recognized; available output is sent after the end-of-line character. When input ends, the TIP positions the cursor to the lefthand margin of the next line on the display.

When normalized mode output is sent to a console, the TIP supports only one output mechanism, the display. Line folding and scrolling are assumed to be performed by the output mechanism when page width (see the PW command in section 5) is reached. When output ends, the TIP ensures that the cursor is positioned at the lefthand margin of the next line on the display. Page waiting is supported.

Transparent Mode

Transparent mode input and output are supported only for the display console. The TIP adds the line protocol frame control characters to the output and removes them from input.

In transparent mode, the application program is responsible for handling all data interpretation, including any embedded IVT command composed of feature control characters. The application program must properly position the cursor at the end of an input or output.

The TIP does not perform an erase write operation at the beginning of transparent mode output. The transmission buffer of the device is therefore not cleared unless the proper control codes are part of the transparent output.

The application program constructs a screen full of protected and unprotected yields and supplies all of the desired attribute-characters and screen-buffer addresses for the fields. The TIP precedes the block of output by SYNC-characters, start-of-text, and escape-char; and attaches ETX, CRC, and PAD at the end. The TIP also translates all downline data from ASCII to EBCDIC and performs SYNC-fill.

A typical start of a field would be:

```
SBA     set-buffer-address x '11'

BA1     buffer-address-1
```

BA2 buffer-address-2

ATT attribute-char

where the attribute-character determines the characteristics (protected, unprotected, intensified, and numeric shift) of the field.

The application program is also expected to insert the cursor at a desired location.

Once transparent output is delivered to a 3270 terminal, the TIP assumes transparent input until a non-transparent downline block is delivered to the terminal.

To protect the integrity of the protocol, the TIP replaces certain downline characters by NULLs. The characters are: SOH, STX, ETX, EOT, ENQ, ACK, NAK, and SYNC.

Once transparent output is delivered, the TIP sends to the host all modified unprotected fields received from the terminal including the SBA and buffer-address-chars of each field. The terminal does not send the attribute-characters back to the TIP.

If the incoming text is larger than one transmission block (256 characters), the TIP sends:

BLK/BLK/.../MSG

so that the application can reproduce a full screen.

Terminal Code Sets and Parity

The 3270 TIP supports terminals operating without parity if they have the IBM EBCDIC code set.

The TIP converts terminal code sets to and from network ASCII code sets. During output to 3270 terminals, the TIP replaces with blanks the ASCII characters that have no EBCDIC equivalents. During input from 3270 terminals, the TIP, again replaces EBCDIC characters that have no ASCII equivalents with blanks (see appendix A).

When the host sends ASCII character codes, the eighth upper bit (bit 7) can be anything because it is ignored when the TIP translates the character. The bit is set as needed for the corresponding EBCDIC character.

When the TIP sends the host ASCII character codes, the eighth bit is always set to 0.

Initial Connection

The initial terminal class, cluster address, terminal address, and device type must be explicitly set by using the network configuration file. There is no automatic recognition performed for 3270 terminals.

The security of a user's login information is ensured by the CCP's maintenance of a security control character. The security character is site-assigned (there is no default) during a CCP build. The character must be: 7-bits, with part of the code set for the terminal, and be specified in ASCII.

The security character is restricted to the following hexadecimal values: 03 through 1F, 21 through 2F, 3A through 3C, 3E through 40, 5B through 60, and 7B through 7E. The value can not be the same as the value for AB, BS, B1, B2, CN, CT, EL, or EB.

Disconnection

The 3270 TIP disconnects a terminal on a dial-up line when:

No devices are connected to a host for two minutes.

The Communication Supervisor requests that the line or terminal be disabled.

A controller or line failure has been detected.

When a console is disconnected from a host, the TIP disconnects any batch devices for which that console is the owning console.

The 3270 TIP does not disconnect terminals on a dedicated line. Input from disabled terminals, or input on disabled hardwired lines, is not requested.

Input Regulation

As mentioned earlier in this section under Data Priorities, the TIP must sometimes suspend input.

If input on a logical line occurs when regulation (in other words, suspension) is needed, the message

WAIT..

is sent. The data transmitted from the terminal is discarded.

Output from the host is delivered to the terminal during input regulation.

The message

REPEAT..

is sent to the terminal as soon as regulation ends and data can again be accepted and stored by the NPU.

If regulation begins after input has been accepted by the Block Interface Program but before the block can be queued for transmission to the host, the message

INPUT DISCARDED..

is sent to the terminal and the block is discarded.

Error Recovery

The 3270 TIP performs error recovery on input data by requesting retransmission of erroneous blocks from the terminal. The TIP detects the following errors:

Cyclic redundancy count errors

Illegal transmission block formats

Unknown responses

Timeout over the communication line

When the TIP receives a negative acknowledgment block (NAK) that indicates an error in reception of downline data, the TIP attempts to retransmit the block 15 times. If the sixteenth attempt to transmit fails, the TIP classifies the cluster as inoperative.

The TIP requests retransmission of a bad upline block 15 times. If the sixteenth attempt to receive fails, the TIP classifies the cluster as inoperative.

If the TIP receives 7 consecutive ENQs, timeouts, or bad responses, it classifies the cluster as inoperative.

If the TIP receives 3 text-terminated-by-ENQs or an ACK0/ACK1 out of sequence, it classifies the cluster as inoperative.

In the case of 31 consecutive errors, the TIP classifies the cluster as inoperative.

Because EBCDIC does not contain a parity bit, the TIP cannot detect a parity error. The flag that becomes the parity error flag bit in the host application block header is not used because data containing parity errors is discarded and replaced by correct data resent from the terminal.

During terminal use, you sometimes need to determine whether an application program is sending garbled information, or whether the terminal hardware is garbling the information it receives. The Terminal Verification Facility (TVF) is provided so that any terminal console can be tested for the second possibility. The current release of TVF provides three tests for such consoles:

A loopback test

A line test

A screen test

TVF can verify the correct operation of any class of terminal console supported by the current release of the network software. Terminals in site-defined terminal classes might not be correctly serviced by TVF.

SELECTION AND CONNECTION

A terminal console connects to TVF in the same manner as it connects to any application program. You can either dial in to a communication line that is configured with TVF as an initial application for the line's logical terminal console device (with full, automatic login of the device), or type in:

TVF cr

in response to any of NVF's APPLICATION prompting lines.

For example, if a terminal of terminal class 2 requires access to TVF via a communication line that is not specially configured for automatic login to TVF, the terminal operator would engage in dialog similar to the following:

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198x.

yy/mm/dd hh.mm.ss HQRTTY1
CDC SUNNYVALE OPERATIONS NOS 2 PSR LEVEL xxx
FAMILY: aaa cr
USER NAME: aaa0127 cr
PASSWORD: pass cr
HQRTTY1 APPLICATION: tvf cr
```

Once connection occurs, TVF displays the banner shown in figure 7-1 on the console. For example, after the login sequence shown above, the following would appear:

```
yy/mm/dd hh.mm.ss
TVF
TERMINAL NAME = HQRTTY1
TERMINAL CLASS = 3/MODEL = 721
LINE WIDTH = 80.
PAGE LENGTH = 0.
NAM VER 1.8 642
..
```

DIALOG AND COMMANDS

The prompting line of two periods is displayed at the terminal following each complete message from TVF and prior to any expected operator entry. After the banner signifying connection appears at the terminal, you can enter any of the commands listed in table 7-1.

TABLE 7-1. LEGAL RESPONSES TO TVF COMMAND PROMPTS

Command	Action Produced by Entry
END cr	Terminates the terminal console's connection to TVF.
HELP cr	Displays a list of all legal TVF commands.
TEST cr	Displays a list of all tests available and the corresponding number that selects each when entered as a command (figure 7-2).
1 cr	Loopback test is entered.
2 cr	Line test is entered.
3 cr	Screen test is entered.

Subsequent dialog and actions depend on the command entered. You must separately enter all commands to TVF. You can enter commands in uppercase or lowercase letters without leading blanks. Except during loopback test operation, the command entered must be one of those appearing in figure 7-2.

If TVF cannot interpret an entry, you see the following message:

ERROR - ENTRY entry

```
yy/mm/dd.    hh.mm.ss.
TVF
TERMINAL NAME = termnam.
TERMINAL CLASS = nn/MODEL tclass
LINE WIDTH = pw.
PAGE LENGTH = pl.
NAM VER 1.x |||
..
```

yy/mm/dd — The last two digits of the calendar year, the month, and the day, taken from the system calendar and indicating the date connection occurred.

hh.mm.ss — The hour, minute, and second connection was completed, taken from the system clock.

termnam — The element name assigned to the terminal device in the network software's network configuration file.

nn — The decimal terminal class.

tclass — The mnemonic identifier of the terminal, corresponding to the terminal class number. This mnemonic has the following values:

nn	tclass
1	M33
2	713
3	721
4	2741
5	M40
6	H2000
7	X364
8	T4014X
9	HASP
10	200UT
11	714X
12	711
13	714
14	HPRE
15	73X
16	2780
17	3780
18	3270
any other number	installation-defined

pw — The decimal page width of the terminal console in characters.

pl — The decimal page length of the terminal console in lines.

x — Variant number of the currently installed TVF version.

||| — PSR level number of the currently installed TVF version.

.. — The standard TVF prompt for terminal operator input.

Figure 7-1. TVF Connection Response Banner

```
PLEASE ENTER TEST LIST DESIRED

    1 - LOOPBACK TEST
    2 - LINE TEST
    3 - SCREEN TEST
```

Figure 7-2. Display of Selectable Tests

where the first five characters of your last type-in appear as the value of entry. This message is followed by another prompting line of periods. Other diagnostic messages that can occur during dialog are described in appendix B.

TEST OPERATIONS

When you enter any of the test selection commands 1, 2, or 3, TVF displays a message indicating that the chosen test is beginning. This message is followed by a prompting line of two periods. You can then enter the character stream to be used in the test. The line and screen tests permit you to enter an empty input line; this entry causes TVF to select the character stream used in the test. TVF uses the 96-character subset of the ASCII character set described in appendix A (excluding the deletion character), transmitted in ascending order of the characters' numerical equivalents.

Terminal input can sometimes be restricted by the TIP because of conditions in the network, as described in section 4. Similarly, output from an application program to a terminal can be temporarily or permanently stopped by network software failure, terminal hardware failure, or lack of terminal buffer space. When one of these output interruptions occurs, TVF interprets the interruption as a terminal failure. TVF stops any test execution in progress and ends its connection with the terminal. If the terminal actually has failed, the network software attempts to disconnect the terminal from the network.

Other conditions can cause TVF to stop test execution without disconnecting the terminal. These conditions include entry of either a user-break-1 or user-break-2 character from the terminal, or online use of a mode 4A card reader. TVF responds to such events by issuing a new two-period prompting line, for selection of another test.

Completion of a loopback or line test produces an informative message. No message is produced at the end of a screen test.

LOOPBACK TEST

The loopback test begins when the following appears on the terminal output device:

```
LOOPBACK TEST BEGINS
..
```

When the TEST BEGINS and period prompts appear, you can enter as many separate lines of text as desired. Each line is returned to the terminal in the same form as entered. If you enter an empty line, TVF returns an empty line and a carriage return operation occurs.

This test returns to the terminal whatever character or characters you enter until you enter the command:

```
ENDL [cr]
```

The ENDL command terminates the loopback test so that you can enter another TVF command. This next command must be entered separately, after the following appears on the terminal output device:

```
TVF TEST COMPLETE
..
```

For example, to cause loopback of a line containing the lowercase letter a and another line containing the lowercase letters bcd, you would enter the number 1 after any period prompting line, then engage in the following dialog:

```
LOOPBACK TEST BEGINS
.. a [cr]
a
bcd [cr]
bcd
ENDL [cr]
TVF TEST COMPLETE
..
```

LINE TEST

The line test begins when the following appears on the console:

```
LINE TEST BEGINS
..
```

This test sends one line pw characters long to the terminal console. The pw value used is the current page width in effect for your terminal console. Unless you change the page width after connecting the terminal console to TVF, the pw value shown in the initial connection banner is used. The line test does not occur if a zero page width is in effect for the console.

You specify the characters sent in the response to the TEST BEGINS and period prompts. You can respond to the prompts by entering any one character or by entering an empty line. If you enter a character, a full line comprised of repeated occurrences of that character is returned to the terminal. If you enter an empty line, TVF sends a full line comprised of the beginning portion of the ASCII 96-character subset (excluding the deletion character) described in appendix A.

As soon as TVF is sure the line has been delivered to the terminal, it sends the following lines:

```
TVF TEST COMPLETE
..
```

TVF then waits for you to select another test.

For example, to display a full line of the lowercase letter a, you would enter a 2 after any period prompting line, then engage in the following dialog:

```
LINE TEST BEGINS
.. a [cr]
aaaaaaaaaaaaaaaaaaa⟩ ⟨aaaaaaaaaaaaa
TVF TEST COMPLETE
..
```

SCREEN TEST

The screen test begins when the following appears on the terminal output device:

```
SCREEN TEST BEGINS
..
```

This test sends one line less than a full screen or page of data to the terminal. You specify the data sent in the response to the TEST BEGINS and period prompts. You can respond to the prompts by entering any one character or by entering an empty line. If you enter a character, the screen is cleared, and a full screen or page comprised of repeated occurrences of that character is returned to the terminal. If you enter an empty line, TVF clears the screen and sends a full screen or page comprised of repeated portions of the ASCII 96-character subset described in appendix A, excluding the deletion character.

Because the test fills the entire screen or page, there is no TEST COMPLETE message or prompt displayed. When the screen or page has been filled, you can enter another command as if a TEST COMPLETE message and prompt had appeared.

When the console is defined as having a page length of zero, TVF transmits a page consisting of either pw occurrences (one line) of the character you enter, or as many lines as needed (each pw characters long) to produce the 96-character ASCII subset (excluding the deletion character). In all other cases, TVF defines a full screen or page of characters as:

$$95 \leq pw \times (pl-1)-1 \leq 2000$$

characters, where pw and pl are the values appearing in the initial connection banner displayed by TVF. If you change the page width or page length value in effect for your console after connecting the console to TVF, the new value is used instead. A zero page width or page length value produces the 95-character minimum display.

For example, to display a full screen of the lowercase letter a and a full screen of characters chosen by TVF at a console with a page length of zero, you would enter a 3 in response to any period prompting line, then engage in the following dialog:

```
SCREEN TEST BEGINS
.. a [cr]
aaaaaaaaaaaaaaaaaaa⟩      ⟨aaaaaaaaaaaaaaaaaaa
3 [cr]
SCREEN TEST BEGINS
.. [cr]
!"#$%& ' ( ) * + , -⟩      ⟨abcdef
ghijklmnopqrstuvwxyz { | } ~
```

In this example, you reenter the screen test by entering a 3 without first receiving a prompting line of periods. You can enter any other valid command after the final line of output, without waiting for a prompting line of periods.

DISCONNECTION

When you want to end connection with TVF, you enter the following command in response to any prompting line of periods:

END cr

TVF provides no response to this command but causes the console to be switched to communicate with NVF. Any further dialog that occurs depends on the way the console has been configured.

CHARACTER DATA INPUT, OUTPUT, AND CENTRAL MEMORY REPRESENTATION

This appendix describes the code and character sets used by the operating system local batch device driver programs, magnetic tape driver programs, and network terminal communication products. This appendix does not describe how other products associate certain graphic or control characters with specific binary code values for collating or syntax processing purposes. The main text of this manual describes such associations that are relevant to the reader.

CHARACTER SETS AND CODE SETS

A character set differs from a code set. A character set is a set of graphic and/or control character symbols. A code set is a numbering system used to represent each character within a character set. Characters exist outside the computer system and communication network; codes are received, stored, retrieved, and transmitted within the computer system and network.

When this manual refers to the ASCII 128-character set or the 7-bit ASCII code set, it is referring to the character set and code set defined as the American National Standard Code for Information Interchange (ASCII, ANSI Standard X3.4-1977). References in this manual to an ASCII character set or an ASCII code set do not necessarily apply to the 128-character, 7-bit ASCII code set.

GRAPHIC AND CONTROL CHARACTERS

A graphic character can be displayed or printed. Examples of graphic characters are the characters A through Z, a blank, and the digits 0 through 9. A control character is not a graphic character; a control character initiates, modifies, or stops a control operation. An example of a control character is the backspace character, which moves the terminal carriage or cursor back one space. Although a control character is not a graphic character, some terminals use a graphic representation for control characters.

CODED AND BINARY CHARACTER DATA

Character codes can be interpreted as coded character data or as binary character data. Coded character data is converted by default from one code set representation to another as it enters or leaves the computer system; for example, data received from a terminal or sent to a magnetic tape unit is converted. Binary character data is not converted as it enters or leaves the system. Character codes are not converted when moved within the system; for example, data transferred to or from mass storage is not converted.

The distinction between coded character data and binary character data is important when reading or punching cards and when reading or writing magnetic tape. Only coded character data can be properly reproduced as characters on a line printer. Only binary character data can properly represent characters on a punched card when the data cannot be stored as display code.

The distinction between binary character data and characters represented by binary data (such as peripheral equipment instruction codes) is also important. Only binary noncharacter data can properly reproduce characters on a plotter.

CHARACTER SET TABLES

The character set tables in this appendix are designed so that the user can find the character represented by a code (such as in a dump) or find the code that represents a character. To find the character represented by a code, the user looks up the code in the column listing the appropriate code set and then finds the character on that line in the column listing the appropriate character set. To find the code that represents a character, the user looks up the character and then finds the code on the same line in the appropriate column.

NETWORK OPERATING SYSTEM

NOS supports the following character sets:

- CDC graphic 64-character set
- CDC graphic 63-character set
- ASCII graphic 64-character set
- ASCII graphic 63-character set
- ASCII graphic 95-character set
- ASCII 128-character set

Each installation must select either a 64-character set or a 63-character set. The differences between the codes of a 63-character set and the codes of a 64-character set are described under Character Set Anomalies. Any reference in this appendix to a 64-character set implies either a 63- or 64-character set unless otherwise stated.

NOS supports the following code sets to represent its character sets in central memory:

- 6-bit display code
- 12-bit ASCII code
- 6/12-bit display code

The 6-bit display code is a set of octal codes from 00 to 77, inclusive.

The 12-bit ASCII code is the ASCII 7-bit code right-justified in a 12-bit byte. The bits are numbered from the right starting with 0; bits 0 through 6 contain the ASCII code, bits 7 through 10 contain zeros, and bit 11 distinguishes the 12-bit ASCII 0000 code from the 12-bit 0000 end-of-line byte. The octal values for the 12-bit codes are 0001 through 0177 and 4000.

The 6/12-bit display code is a combination of 6-bit codes and 12-bit codes. The octal values for the 6-bit codes are 00 through 77, excluding 74 and 76. (The interpretation of the 00 and 63 codes is described under Character Set Anomalies in this appendix.) The octal 12-bit codes begin with either 74 or 76 and are followed by a 6-bit code. Thus, 74 and 76 are escape codes and are never used as 6-bit codes within the 6/12-bit display code set. The octal values of the 12-bit codes are: 7401, 7402, 7404, 7407, and 7601 through 7677. The other 12-bit codes, 74xx and 7600, are undefined.

CHARACTER SET ANOMALIES

The operating system input/output software and some products interpret two codes differently when the installation selects a 63-character set rather than a 64-character set. If a site uses a 63-character set: the colon (:) graphic character is always represented by a 6-bit display code value of 63 octal; display code 00 is undefined (it has no associated graphic or punched card code); the percent (%) graphic does not exist, and translations produce a space (55 octal).

However, if the site uses a 64-character set, output of an octal 7404 6/12-bit display code or a 6-bit display code value of 00 produces a colon. In ASCII mode, a colon can be input only as a 7404 6/12-bit display code. Undefined 6/12-bit display codes in output files produce unpredictable results and should be avoided.

Two consecutive 6-bit display code values of 00 can be confused with the 12-bit 0000 end-of-line byte and should be avoided.

Translation of 7-bit or 12-bit ASCII to 6-bit display code causes character folding from the 128-character ASCII set to the 63- or 64-character ASCII subset, with the special character substitutions shown in figure A-1.

INTERACTIVE TERMINAL USERS

NOS supports display consoles and teletypewriters that use code sets other than 7-bit ASCII codes for communication or use graphics other than those defined in an ASCII character set. Data exchanged with such terminals is translated as described under Terminal Transmission Modes in this appendix. The following description applies only to terminals that use 7-bit ASCII codes and the ASCII character set.

ASCII Data Exchange Modes

Table A-1 shows the character sets and code sets available to an Interactive Facility (IAF) user. Table A-2 shows the octal and hexadecimal 7-bit ASCII code for each ASCII character, and can be used to convert codes from octal to hexadecimal. (Certain Terminal Interface Program commands require hexadecimal specification of a 7-bit ASCII code.)

IAF supports both normalized mode and transparent mode transmissions through the network. These transmission modes are described under Terminal Transmission Modes in this appendix. Refer to the NOS Version 2 Reference Set, Volume 3 System Commands, for additional information.

IAF treats normalized mode transmissions as coded character data; IAF converts these transmissions to or from either 6-bit or 6/12-bit display code.

IAF treats transparent mode transmissions as binary character data. Transparent mode input or output uses 12-bit bytes, with bit 11 always set to 1; for ASCII terminals, transparent mode input and output occurs in the 12-bit ASCII code shown in table A-1, but the leftmost digit is 4 instead of 0.

When the NORMAL command is in effect, IAF assumes that the ASCII graphic 64-character set is used and translates all input and output to or from display code. When the ASCII command is in effect, IAF assumes that the ASCII 128-character set is used and translates all input and output to or from 6/12-bit display code.

The IAF user can convert a 6/12-bit display code file to a 12-bit ASCII code file using the NOS FCOPY control statement. The resulting 12-bit ASCII file can be routed to a line printer but the file cannot be output through IAF.

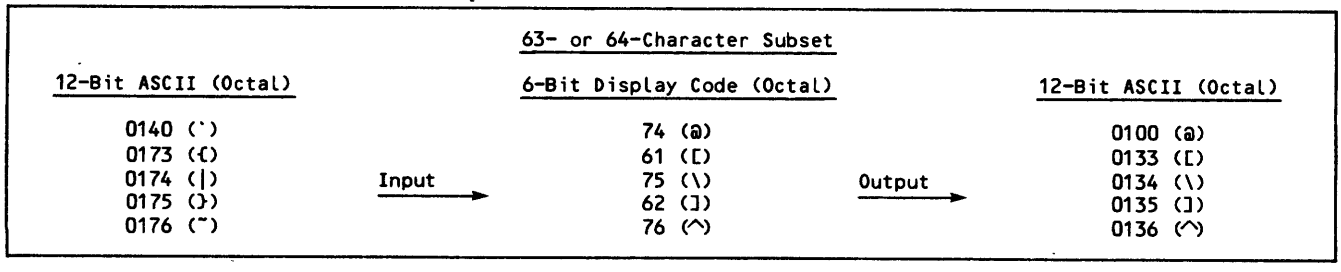


Figure A-1. ASCII Character Folding

Terminal Transmission Modes

Coded character data can be exchanged with a conversational terminal in two transmission modes. These two modes, normalized mode and transparent mode, correspond to the types of character code editing and translation performed by the network software during input and output operations.

The terminal operator can change the input transmission mode using a terminal definition command (sometimes called a Terminal Interface Program command). The application program providing the terminal facility service can change the input or output transmission mode.

Normalized Mode Transmissions

Normalized mode is the initial and default mode used for both input and output transmissions. The network software translates normalized mode data to or from the transmission code used by the terminal into or from the 7-bit ASCII code shown in table A-2. (Tables A-1 and A-3 through A-7 are provided for use while coding an application program to run under the operating system; they do not describe character transmissions through the network.) Translation of a specific terminal transmission code to or from a specific 7-bit ASCII code depends on the terminal class in which the network software places the terminal.

The following paragraphs summarize the general case for normalized mode data code translations. This generalized description uses table A-2.

The reader can extend this generalized description by using the other tables to determine character set mapping for functions initiated from a terminal. For example, the description under Terminal Output Character Sets can be used to predict whether a lowercase ASCII character stored in 6/12-bit display code can appear on an EBCDIC or external BCD terminal; if an ASCII character passes through the network represented in 7-bit ASCII as character mode data, it probably can be represented on an EBCDIC terminal, but it is always transformed to an uppercase character on a mode 4A ASCII terminal.

Table A-2 contains the ASCII 128-character set supported by the network software. The ASCII 96-character subset in the rightmost six columns minus the deletion character (DEL) comprises the graphic 95-character subset; the DEL is not a graphic character, although some terminals graphically represent it. The graphic 64-character subset comprises the middle four columns. Only the characters in this 64-character subset have 6-bit display code equivalents.

Terminals that support an ASCII graphic 64-character subset actually use a subset of up to 96 characters, consisting of the graphic 64-character subset and the control characters of columns 1 and 2; often, the DEL character in column 7 is included. Terminals that support an ASCII graphic 95-character or 96-character subset actually might use all 128 characters.

The hexadecimal value of the 7-bit code for each character in table A-2 consists of the character's column number in the table, followed by its row number. For example, N is in row E of column 4, so its hexadecimal value is 4E. The octal value for the code when it is right-justified in an 8-bit byte appears beneath the character graphic or mnemonic. The binary value of the code consists of the bit values shown, placed in the order given by the subscripts for the letter b; for example, N is 1001110.

Tables A-8 through A-24 show the normalized mode translations performed for each terminal class. The parity shown in the terminal transmission codes is the parity used as a default for the terminal class. The parity setting actually used by a terminal can be identified to the network software through a TIP command.

Tables A-8 through A-24 contain the graphic and control characters associated with the transmission codes used by the terminal because of the terminal class and code set in use. The network ASCII graphic and control characters shown are those of the standard ASCII character set associated with the ASCII transmission codes of table A-2.

Terminal Output Character Subsets -- Although the network supports the ASCII 128-character set, some terminals restrict output to a smaller character set. This restriction is supported by replacing the control characters in columns 0 and 1 of table A-2 with blanks to produce the ASCII graphic 95-character subset, and replacing the characters in columns 6 and 7 with the corresponding characters from columns 4 and 5, respectively, to produce the ASCII graphic 64-character subset.

Terminal Input Character Subsets and Supersets -- Although the network supports the ASCII 128-character set, some terminals restrict input to a smaller character set or permit input of a larger character set. A character input from a device using a character set other than ASCII is converted to an equivalent ASCII character; terminal characters without ASCII character equivalents are represented by the ASCII code for a space.

Site-written terminal-servicing facility programs can also cause input or output character replacement, conversion, or deletion by exchanging data with the network in 6-bit display code.

Input Restrictions -- The network software automatically deletes codes associated with terminal communication protocols or terminal hardware functions. These codes usually represent the cancel, backspace, linefeed, carriage return, and deletion characters. If paper tape support is requested, the device control 3 code also is deleted. Some of these code deletions can be suppressed by using the full-ASCII and special editing options (refer to the FA and SE terminal definition parameters in the NOS Version 2 Reference Set, Volume 3, System Commands).

Output Restrictions -- All codes sent by an application program are transmitted to the terminal. However, the 12-bit ASCII code 0037 (octal), the 6/12-bit display code 7677 (octal), and the 7-bit ASCII code 1F (hexadecimal) should be avoided in character mode output. The network software interprets the unit separator character represented by these codes as an end-of-line indicator. The processing of application program-supplied unit separators causes incorrect formatting of output and can cause loss of other output characters.

Input Parity Processing -- The network software does not preserve the parity of the terminal transmission code in the corresponding ASCII code. An ASCII code received by the terminal-servicing facility program always contains zero as its eighth bit.

Output Parity Processing -- The network software provides the parity bit setting appropriate for the terminal being serviced, even when the software is translating from ASCII character codes with zero parity bit settings.

Transparent Mode Transmissions

Transparent mode is selected separately for input and output transmissions.

During transparent mode input, the parity bit is stripped from each terminal transmission code (unless the N or I parity option has been selected by a terminal definition command), and the transmission code is placed in an 8-bit byte without translation to 7-bit ASCII code. Line transmission protocol characters are deleted from mode 4 terminal input. When the 8-bit bytes arrive in the host computer, a terminal servicing facility program can right-justify the bytes within a 12-bit byte.

During transparent mode output, processing similar to that performed for input occurs. When the host computer transmits 12-bit bytes, the leftmost 4 bits (bits 11 through 8) are discarded. The code in each 8-bit byte received by the network software is not translated. The parity bit appropriate for the terminal class is altered as indicated by the parity option in effect for the terminal. The codes are then transmitted to the terminal in bytes of a length appropriate for the terminal class. Line transmission protocol characters are inserted into mode 4 terminal output.

LOCAL BATCH USERS

Table A-3 lists the CDC graphic 64-character set, the ASCII graphic 64-character set, and the ASCII graphic 95-character set available on local batch devices. This table also lists the code sets and card keypunch codes (026 and 029) that represent the characters.

The 64-character sets use 6-bit display code as their code set; the 95-character set uses 12-bit ASCII code. The 95-character set is composed of all the characters in the ASCII 128-character set that can be printed at a line printer (refer to Line Printer Output). Only 12-bit ASCII code files can be printed using the graphic ASCII 95-character set. The 95-character set is represented by the octal 12-bit ASCII codes 0040 through 0176. An octal 12-bit ASCII code outside of the range 0040 through 0176 represents an unprintable character.

To print a 6/12-bit display code file, the user must convert the file to 12-bit ASCII code. The NOS FCOPY control statement is used for this conversion.

Line Printer Output

The printer train used on the line printer to which a file is sent determines which batch character set is printed. The following CDC print trains match the batch character sets in table A-3:

Character Set	Print Train
CDC graphic 64-character set	596-1
ASCII graphic 64-character set	596-5
ASCII graphic 95-character set	596-6

The characters of the default 596-1 print train are listed in the table A-3 column labeled CDC Graphic (64-Character Set); the 596-5 print train characters are listed in the table A-3 column labeled ASCII Graphic (64-Character Set); and the 596-6 print train characters are listed in the table A-3 column labeled ASCII Graphic (95-Character Set).

If an unprintable character exists in a line, NOS marks the condition by printing the number sign (#) in the first printable column of the line. A space replaces the unprintable character within the line.

When a transmission error occurs during the printing of a line, NOS makes up to five attempts to reprint the line. The CDC graphic print train prints a concatenation symbol (↱) in the first column of the repeated line following a line containing errors. The ASCII print trains print an underline (_) instead of the concatenation symbol.

After the fifth attempt, the setting of sense switch one for the batch input and output control point determines further processing. NOS either rewinds the file and returns it to the print queue, or ignores the transmission errors.

Punched Card Input and Output

A character represented by multiple punches in a single column has its punch pattern identified by numbers and hyphens. For example, the punches representing an exclamation point are identified as 11-0; this notation means both rows 11 and 0 are punched in the same column.

A multiple punch pattern that represents something other than a character is identified by numbers and slashes. For example, the punches representing the end of an input file are identified as 6/7/8/9; this notation means rows 6 through 9 are punched in the same column.

Coded character data is exchanged with card readers or card punches according to the translations shown in table A-3. As indicated in the table, other card keypunch codes are available for input of the ASCII and CDC characters [and]. NOS cannot read or punch the 95-character set as coded character data.

Each site chooses either 026 or 029 as its default keypunch code. NOS begins reading an input deck in the default code (regardless of the character set

in use). The user can specify the alternate keypunch code by punching a 26 or 29 in columns 79 and 80 of any job card, 6/7/9 card, or 7/8/9 card. The specified translation continues throughout the job unless the alternate keypunch code translation is specified on a subsequent 6/7/9 or 7/8/9 card.

A 5/7/9 card with a punch in column 1 changes keypunch code translation if the card is read immediately before or after a 7/8/9 card. A space (no punch) in column 2 indicates 026 translation mode; a 9 punch in column 2 indicates 029 translation mode. The specified translation remains in effect until a similar 5/7/9 card or a 7/8/9 card is encountered, or the job ends.

The 5/7/9 card also allows literal input when 4/5/6/7/8/9 is punched in column 2. Literal input can be used to read 80-column binary character data within a punched card deck of coded character data.

Literal cards are stored with each column represented in a 12-bit byte (a row 12 punch is represented by a 1 in bit 11, row 11 by a 1 in bit 10, row 0 by a 1 in bit 9, and rows 1 through 9 by 1's in bits 8 through 0 of the byte), using 16 central memory words per card. Literal input cards are read until another 5/7/9 card with 4/5/6/7/8/9 punched in column 2 is read. The next card can specify a new conversion mode.

If the card following the 5/7/9, 6/7/9, or 7/8/9 card has a 7 and a 9 punched in column 1, the section of the job deck following it contains system binary cards (as described in the NOS Version 2 Reference Set, Volume 3, System Commands).

REMOTE BATCH USERS

Remote batch console input and output is restricted to character mode transmission. Character mode is described under Terminal Transmission Modes in this appendix.

The abilities to select alternate keypunch code translations, to read binary cards, to output plotter files, and to print lowercase characters depend upon the remote terminal equipment. Remote batch terminal support under NOS is described in the Remote Batch Facility (RBF) reference manual.

MAGNETIC TAPE USERS

The character and code sets used for reading and writing magnetic tapes depend on whether coded or binary data is read or written and on whether the tape is 7-track or 9-track.

Coded Data Exchanges

Coded character data to be copied from mass storage to magnetic tape is assumed to be stored in a 63- or 64-character 6-bit display code. The operating system magnetic tape driver program converts the data to 6-bit external BCD code when writing a coded 7-track tape and to 7-bit ASCII or 8-bit EBCDIC code (as specified on the tape assignment statement) when writing a coded 9-track tape.

Coded character data copied to mass storage from magnetic tape is stored in a 63- or 64-character 6-bit display code. The operating system magnetic tape driver program converts the data from 6-bit external BCD code when reading a coded 7-track tape and from 7-bit ASCII or 8-bit EBCDIC code (as specified on the tape assignment statement) when reading a coded 9-track tape.

To read and write lowercase character 7-bit ASCII or 8-bit EBCDIC codes or to read and write control codes, the user must assign a 7-track or 9-track tape in binary mode.

Seven-Track Tape Input and Output

Table A-4 shows the code and character set conversions between 6-bit external BCD and 6-bit display code for 7-track tapes. Because only 63 characters can be represented in 7-track even parity, one of the 64 display codes is lost in conversion to and from external BCD code.

Figure A-2 shows the differences in 7-track tape conversion that depend on whether the system uses the 63-character or 64-character set. The ASCII character for the specified character code is shown in parentheses. The output arrows show how the 6-bit display code changes when it is written on tape in external BCD. The input arrows show how the external BCD code changes when the tape is read and converted to display code.

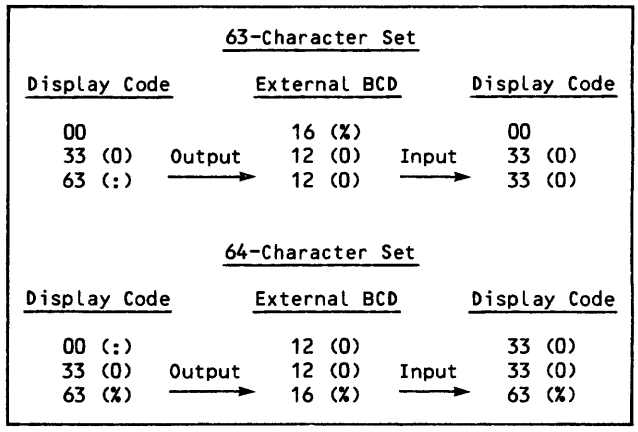


Figure A-2. Magnetic Tape Code Conversions

Nine-Track Tape Input and Output

Table A-5 lists the conversions between the 7-bit ASCII code used on the tape and the 6-bit display code used within the system. Table A-6 lists the conversions between the 8-bit EBCDIC code used on the tape and the 6-bit display code used within the system.

When an ASCII or EBCDIC code representing a lowercase character is read from a 9-track magnetic tape, it is converted to its uppercase character

6-bit display code equivalent. Any EBCDIC code not listed in table A-6 is converted to display code 55 (octal) and becomes a space. Any code between 80 (hexadecimal) and FF (hexadecimal) read from an ASCII tape is converted to display code 00.

Binary Character Data Exchanges

Binary character data exchanged between central memory files and magnetic tape is transferred as a string of bytes without conversion of the byte contents. The grouping of the bytes and the number of bits in each byte depend on whether 7-track or 9-track tape is being used.

Seven-Track Tape Input and Output

Each binary data character code written to or read from 7-track magnetic tape is assumed to be stored in a 6-bit byte, such as the system uses for 63- or 64-character 6-bit display code. Seven-bit ASCII and 8-bit EBCDIC codes can only be read from or written to 7-track magnetic tape as binary character data if each code is stored within a 12-bit byte as if it were two character codes.

Nine-Track Tape Input and Output

Each binary data character code exchanged between central memory files and 9-track magnetic tape is assumed to be stored in an 8-bit or 12-bit byte. During such binary data transfers, the 6/12-bit display codes and 12-bit ASCII codes shown in table A-1, the 7-bit ASCII codes shown in table A-2, or or the 8-bit hexadecimal EBCDIC codes shown in table A-7 can be read or written. The 7-bit ASCII codes and 8-bit EBCDIC codes can be exchanged either in an unformatted form or right-justified within a zero-filled 12-bit byte of memory.

When 9-track tape is written, every pair of 12-bit memory bytes becomes three 8-bit tape bytes; when 9-track tape is read, every three 8-bit tape bytes become a pair of 12-bit memory bytes. Because of the 12-bit byte pairs, codes not packed into 12-bit bytes are exchanged in their unpacked form, while codes packed in 12-bit bytes are exchanged in packed form.

When an odd number of central memory words is read or written, the lower four bits of the last 8-bit byte (bits 0 through 3 of the last word) are not used. For example, three central memory words are written on tape as 22 8-bit bytes (7.5 pairs of 12-bit bytes) and the remaining four bits are ignored.

CODE CONVERSION AIDS

Table A-7 contains the octal values of each 8-bit EBCDIC code right-justified in a 12-bit byte with zero fill. This 12-bit EBCDIC code can be produced or read using the FORM and 8-Bit Subroutines utilities.

TABLE A-1. INTERACTIVE TERMINAL CHARACTER SETS

Character Sets		Code Sets		
ASCII Graphic (64-Character Set)	ASCII Character (128-Character Set)	Octal 6-Bit Display Code	Octal 6/12-Bit Display Code†	Octal 12-Bit ASCII Code
: colon††		00††		
A	A	01	01	0101
B	B	02	02	0102
C	C	03	03	0103
D	D	04	04	0104
E	E	05	05	0105
F	F	06	06	0106
G	G	07	07	0107
H	H	10	10	0110
I	I	11	11	0111
J	J	12	12	0112
K	K	13	13	0113
L	L	14	14	0114
M	M	15	15	0115
N	N	16	16	0116
O	O	17	17	0117
P	P	20	20	0120
Q	Q	21	21	0121
R	R	22	22	0122
S	S	23	23	0123
T	T	24	24	0124
U	U	25	25	0125
V	V	26	26	0126
W	W	27	27	0127
X	X	30	30	0130
Y	Y	31	31	0131
Z	Z	32	32	0132
0	0	33	33	0060
1	1	34	34	0061
2	2	35	35	0062
3	3	36	36	0063
4	4	37	37	0064
5	5	40	40	0065
6	6	41	41	0066
7	7	42	42	0067
8	8	43	43	0070
9	9	44	44	0071
+ plus	+ plus	45	45	0053
- hyphen (minus)	- hyphen (minus)	46	46	0055
* asterisk	* asterisk	47	47	0052
/ slant	/ slant	50	50	0057
(opening parenthesis	(opening parenthesis	51	51	0050
) closing parenthesis	) closing parenthesis	52	52	0051
$ dollar sign	$ dollar sign	53	53	0044
= equals	= equals	54	54	0075
space	space	55	55	0040
, comma	, comma	56	56	0054
. period	. period	57	57	0056
# number sign	# number sign	60	60	0043
[opening bracket	[opening bracket	61	61	0133
] closing bracket	] closing bracket	62	62	0135
% percent sign††	% percent sign††	63††	63††	0045
" quotation mark	" quotation mark	64	64	0042
_ underline	_ underline	65	65	0137
! exclamation point	! exclamation point	66	66	0041
& ampersand	& ampersand	67	67	0046
' apostrophe	' apostrophe	70	70	0047
? question mark	? question mark	71	71	0077

TABLE A-1. INTERACTIVE TERMINAL CHARACTER SETS (Contd)

Character Sets		Code Sets		
ASCII Graphic (64-Character Set)	ASCII Character (128-Character Set)	Octal 6-Bit Display Code	Octal 6/12-Bit Display Code†	Octal 12-Bit ASCII Code
< less than	< less than	72	72	0074
> greater than	> greater than	73	73	0076
@ commmercial at	@ commercial at	74††	7401††	0100
\ reverse slant	\ reverse slant	75	75	0134
^ circumflex		76		
; semicolon	; semicolon	77	77	0073
	^ circumflex	76††	7402	0136
	: colon††	74††	7404††	0072
	` grave accent		7407	0140
	a		7601	0141
	b		7602	0142
	c		7603	0143
	d		7604	0144
	e		7605	0145
	f		7606	0146
	g		7607	0147
	h		7610	0150
	i		7611	0151
	j		7612	0152
	k		7613	0153
	l		7614	0154
	m		7615	0155
	n		7616	0156
	o		7617	0157
	p		7620	0160
	q		7621	0161
	r		7622	0162
	s		7623	0163
	t		7624	0164
	u		7625	0165
	v		7626	0166
	w		7627	0167
	x		7630	0170
	y		7631	0171
	z		7632	0172
	{ opening brace	61††	7633	0173
	| vertical line	75††	7634	0174
	} closing brace	62††	7635	0175
	~ tilde	76††	7636	0176
	NUL		7640	4000
	SOH		7641	0001
	STX		7642	0002
	ETX		7643	0003
	EOT		7644	0004
	ENQ		7645	0005
	ACK		7646	0006
	BEL		7647	0007
	BS		7650	0010
	HT		7651	0011
	LF		7652	0012
	VT		7653	0013
	FF		7654	0014
	CR		7655	0015
	SO		7656	0016
	SI		7657	0017
	DEL		7637	0177
	DLE		7660	0020

TABLE A-1. INTERACTIVE TERMINAL CHARACTER SETS (Contd)

Character Sets		Code Sets		
ASCII Graphic (64-Character Set)	ASCII Character (128-Character Set)	Octal 6-Bit Display Code	Octal 6/12-Bit Display Code†	Octal 12-Bit ASCII Code
	DC1		7661	0021
	DC2		7662	0022
	DC3		7663	0023
	DC4		7664	0024
	NAK		7665	0025
	SYN		7666	0026
	ETB		7667	0027
	CAN		7670	0030
	EM		7671	0031
	SUB		7672	0032
	ESC		7673	0033
	FS		7674	0034
	GS		7675	0035
	RS		7676	0036
	US		7677	0037

†Available only on NOS.

††Character or code interpretation depends on context. Refer to Character Set Anomalies in the text.

TABLE A-2. 7-BIT ASCII CODE AND CHARACTER SETS

128-Character Set

96-Character Subset

Graphic 64-Character Subset

Bits b_7 b_6 b_5 b_4 b_3 b_2 b_1	Row / Column	0 (0 0 0)	1 (0 0 1)	2 (0 1 0)	3 (0 1 1)	4 (1 0 0)	5 (1 0 1)	6 (1 1 0)	7 (1 1 1)
0 0 0 0	0	NUL 000	DLE 020	SP 040	0 060	@ 100	P 120	` 140	p 160
0 0 0 1	1	SOH 001	DC1 021	! 041	1 061	A 101	Q 121	a 141	q 161
0 0 1 0	2	STX 002	DC2 022	" 042	2 062	B 102	R 122	b 142	r 162
0 0 1 1	3	ETX 003	DC3 023	# 043	3 063	C 103	S 123	c 143	s 163
0 1 0 0	4	EOT 004	DC4 024	$ 044	4 064	D 104	T 124	d 144	t 164
0 1 0 1	5	ENQ 005	NAK 025	% 045	5 065	E 105	U 125	e 145	u 165
0 1 1 0	6	ACK 006	SYN 026	& 046	6 066	F 106	V 126	f 146	v 166
0 1 1 1	7	BEL 007	ETB 027	' 047	7 067	G 107	W 127	g 147	w 167
1 0 0 0	8	BS 010	CAN 030	(050	8 070	H 110	X 130	h 150	x 170
1 0 0 1	9	HT 011	EM 031	) 051	9 071	I 111	Y 131	i 151	y 171
1 0 1 0	A	LF 012	SUB 032	* 052	: 072	J 112	Z 132	j 152	z 172
1 0 1 1	B	VT 013	ESC 033	+ 053	; 073	K 113	[133	k 153	{ 173
1 1 0 0	C	FF 014	FS 034	, 054	< 074	L 114	\ 134	l 154	| 174
1 1 0 1	D	CR 015	GS 035	- 055	= 075	M 115	] 135	m 155	} 175
1 1 1 0	E	SO 016	RS 036	. 056	> 076	N 116	^ 136	n 156	~ 176
1 1 1 1	F	SI 017	US 037	/ 057	? 077	O 117	_ 137	o 157	DEL† 177

†The graphic 95-character subset does not include DEL; refer to Terminal Transmission Modes in the text.

LEGEND:

Numbers under characters are the octal values for the 7-bit character codes used within the network.

TABLE A-3. LOCAL BATCH DEVICE CHARACTER SETS

Character Sets			Code Sets			Card Keypunch Code	
CDC Graphic (64-Character Set)	ASCII Graphic (64-Character Set)	ASCII Graphic (95-Character Set)	Octal 6-Bit Display Code	Octal 6/12-Bit Display Code†	Octal 12-Bit ASCII Code	026	029
: colon††	: colon††		00††			8-2	8-2
A	A	A	01	01	0101	12-1	12-1
B	B	B	02	02	0102	12-2	12-2
C	C	C	03	03	0103	12-3	12-3
D	D	D	04	04	0104	12-4	12-4
E	E	E	05	05	0105	12-5	12-5
F	F	F	06	06	0106	12-6	12-6
G	G	G	07	07	0107	12-7	12-7
H	H	H	10	10	0110	12-8	12-8
I	I	I	11	11	0111	12-9	12-9
J	J	J	12	12	0112	11-1	11-1
K	K	K	13	13	0113	11-2	11-2
L	L	L	14	14	0114	11-3	11-3
M	M	M	15	15	0115	11-4	11-4
N	N	N	16	16	0116	11-5	11-5
O	O	O	17	17	0117	11-6	11-6
P	P	P	20	20	0120	11-7	11-7
Q	Q	Q	21	21	0121	11-8	11-8
R	R	R	22	22	0122	11-9	11-9
S	S	S	23	23	0123	0-2	0-2
T	T	T	24	24	0124	0-3	0-3
U	U	U	25	25	0125	0-4	0-4
V	V	V	26	26	0126	0-5	0-5
W	W	W	27	27	0127	0-6	0-6
X	X	X	30	30	0130	0-7	0-7
Y	Y	Y	31	31	0131	0-8	0-8
Z	Z	Z	32	32	0132	0-9	0-9
0	0	0	33	33	0060	0	0
1	1	1	34	34	0061	1	1
2	2	2	35	35	0062	2	2
3	3	3	36	36	0063	3	3
4	4	4	37	37	0064	4	4
5	5	5	40	40	0065	5	5
6	6	6	41	41	0066	6	6
7	7	7	42	42	0067	7	7
8	8	8	43	43	0070	8	8
9	9	9	44	44	0071	9	9
+ plus	+ plus	+ plus	45	45	0053	12	12-8-6
- hyphen (minus)	- hyphen (minus)	- hyphen (minus)	46	46	0055	11	11
* asterisk	* asterisk	* asterisk	47	47	0052	11-8-4	11-8-4
/ slant	/ slant	/ slant	50	50	0057	0-1	0-1
(open. paren.	(open. paren.	(open. paren.	51	51	0050	0-8-4	12-8-5
) clos. paren.	) clos. paren.	) clos. paren.	52	52	0051	12-8-4	11-8-5
$ dollar sign	$ dollar sign	$ dollar sign	53	53	0044	11-8-3	11-8-3
= equals	= equals	= equals	54	54	0075	8-3	8-6
space	space	space	55	55	0040	no punch	no punch
, comma	, comma	, comma	56	56	0054	0-8-3	0-8-3
. period	. period	. period	57	57	0056	12-8-3	12-8-3
≡ equivalence	# number sign	# number sign	60	60	0043	0-8-6	8-3
[open. bracket	[open. bracket	[open. bracket	61	61	0133	8-7	12-8-2 or 12-0†††
] clos. bracket	] clos. bracket	] clos. bracket	62	62	0135	0-8-2	11-8-2 or 11-0†††
% percent sign††	% percent sign††	% percent sign††	63††	63††	0045	8-6	0-8-4

TABLE A-3. LOCAL BATCH DEVICE CHARACTER SETS (Contd)

Character Sets			Code Sets			Card Keypunch Code	
CDC Graphic (64-Character Set)	ASCII Graphic (64-Character Set)	ASCII Graphic (95-Character Set)	Octal 6-Bit Display Code	Octal 6/12-Bit Display Code†	Octal 12-Bit ASCII Code	026	029
≠ not equals	" quotation mark	" quotation mark	64	64	0042	8-4	8-7
↦ concatenation.	_ underline	_ underline	65	65	0137	0-8-5	0-8-5
∨ logical OR	! exclamation pt.	! exclamation pt.	66	66	0041	11-0	12-8-7
						or 11-8-2§	or 11-0§
∧ logical AND	& ampersand	& ampersand	67	67	0046	0-8-7	12
↑ superscript	' apostrophe	' apostrophe	70	70	0047	11-8-5	8-5
↓ subscript	? question mark	? question mark	71	71	0077	11-8-6	0-8-7
< less than	< less than	< less than	72	72	0074	12-0	12-8-4
						or 12-8-2§	or 12-0§
> greater than	> greater than	> greater than	73	73	0076	11-8-7	0-8-6
≤ less/equal	@ commercial at	@ commercial at	74††	7401††	0100	8-5	8-4
≥ greater/equal	\ reverse slant	\ reverse slant	75	75	0134	12-8-5	0-8-2
¬ logical NOT	^ circumflex		76			12-8-6	11-8-7
; semicolon	; semicolon	; semicolon	77	77	0073	12-8-7	11-8-6
		^ circumflex	76††	7402	0136		
		: colon††		7404††	0072		
		` grave accent	74††	7407	0140		
		a		7601	0141		
		b		7602	0142		
		c		7603	0143		
		d		7604	0144		
		e		7605	0145		
		f		7606	0146		
		g		7607	0147		
		h		7610	0150		
		i		7611	0151		
		j		7612	0152		
		k		7613	0153		
		l		7614	0154		
		m		7615	0155		
		n		7616	0156		
		o		7617	0157		
		p		7620	0160		
		q		7621	0161		
		r		7622	0162		
		s		7623	0163		
		t		7624	0164		
		u		7625	0165		
		v		7626	0166		
		w		7627	0167		
		x		7630	0170		
		y		7631	0171		
		z		7632	0172		
		{ open. brace	61††	7633	0173		
		| vertical line	75††	7634	0174		
		} clos. brace	62††	7635	0175		
		~ tilde	76†††	7636	0176		

†Available only on NOS.

††Character or code interpretation depends on context. Refer to Character Set Anomalies in the text.

†††Available for input only, on NOS.

§Available for input only, on NOS/BE or SCOPE 2.

TABLE A-4. 7-TRACK CODED TAPE CONVERSIONS

External BCD	ASCII Character	Octal 6-Bit Display Code	External BCD	ASCII Character	Octal 6-Bit Display Code
01	1	34	40	- hyphen (minus)	46
02	2	35	41	J	12
03	3	36	42	K	13
04	4	37	43	L	14
05	5	40	44	M	15
06	6	41	45	N	16
07	7	42	46	O	17
10	8	43	47	P	20
11	9	44	50	Q	21
12†	0	33	51	R	22
13	= equals	54	52	! exclamation point	66
14	" quotation mark	64	53	$ dollar sign	53
15	@ commercial at	74	54	* asterisk	47
16†	% percent sign	63	55	' apostrophe	70
17	[opening bracket	61	56	? question mark	71
20	space	55	57	> greater than	73
21	/ slant	50	60	+ plus	45
22	S	23	61	A	01
23	T	24	62	B	02
24	U	25	63	C	03
25	V	26	64	D	04
26	W	27	65	E	05
27	X	30	66	F	06
30	Y	31	67	G	07
31	Z	32	70	H	10
32	] closing bracket	62	71	I	11
33	, comma	56	72	< less than	72
34	(opening parenthesis	51	73	. period	57
35	_ underline	65	74	) closing parenthesis	52
36	# number sign	60	75	\ reverse slant	75
37	& ampersand	67	76	^ caret	76
			77	; semicolon	77

†As the text explains, conversion of these codes depends on whether the tape is read or written.

TABLE A-5. ASCII 9-TRACK CODED TAPE CONVERSION

ASCII				6-Bit Display Code†††	
Code Conversion†		Character and Code Conversion††			
Code (Hex)	Character	Code (Hex)	Character	ASCII Character	Code (Octal)
20	space	00	NUL	space	55
21	! exclamation point	7D	} closing brace	! exclamation point	66
22	" quotation mark	02	STX	" quotation mark	64
23	# number sign	03	ETX	# number sign	60
24	$ dollar sign	04	EOT	$ dollar sign	53
25	% percent sign§	05	ENQ	% percent sign§	63§
26	& ampersand	06	ACK	& ampersand	67
27	' apostrophe	07	BEL	' apostrophe	70
28	(opening parenthesis	08	BS	(opening parenthesis	51
29	) closing parenthesis	09	HT	) closing parenthesis	52
2A	* asterisk	0A	LF	* asterisk	47
2B	+ plus	0B	VT	+ plus	45
2C	, comma	0C	FF	, comma	56
2D	- hyphen (minus)	0D	CR	- hyphen (minus)	46
2E	. period	0E	SO	. period	57
2F	/ slant	0F	SI	/ slant	50
30	0	10	DLE	0	33
31	1	11	DC1	1	34
32	2	12	DC2	2	35
33	3	13	DC3	3	36
34	4	14	DC4	4	37
35	5	15	NAK	5	40
36	6	16	SYN	6	41
37	7	17	ETB	7	42
38	8	18	CAN	8	43
39	9	19	EM	9	44
3A	: colon§	1A	SUB	: colon§	00§
3B	; semicolon	1B	ESC	; semicolon	77
3C	< less than	7B	{ opening brace	< less than	72
3D	= equals	1D	GS	= equals	54
3E	> greater than	1E	RS	> greater than	73
3F	? question mark	1F	US	? question mark	71
40	@ commercial at	60	` grave accent	@ commercial at	74
41	A	61	a	A	01
42	B	62	b	B	02
43	C	63	c	C	03
44	D	64	d	D	04
45	E	65	e	E	05
46	F	66	f	F	06

TABLE A-5. ASCII 9-TRACK CODED TAPE CONVERSION (Contd)

ASCII				6-Bit Display Code†††	
Code Conversion†		Character and Code Conversion††			
Code (Hex)	Character	Code (Hex)	Character	ASCII Character	Code (Octal)
47	G	67	g	G	07
48	H	68	h	H	10
49	I	69	i	I	11
4A	J	6A	j	J	12
4B	K	6B	k	K	13
4C	L	6C	l	L	14
4D	M	6D	m	M	15
4E	N	6E	n	N	16
4F	O	6F	o	O	17
50	P	70	p	P	20
51	Q	71	q	Q	21
52	R	72	r	R	22
53	S	73	s	S	23
54	T	74	t	T	24
55	U	75	u	U	25
56	V	76	v	V	26
57	W	77	w	W	27
58	X	78	x	X	30
59	Y	79	y	Y	31
5A	Z	7A	z	Z	32
5B	[opening bracket	1C	FS	[opening bracket	61
5C	\ reverse slant	7C	| vertical line	\ reverse slant	75
5D	] closing bracket	01	SOH	] closing bracket	62
5E	^ caret	7E	~ tilde	^ caret	76
5F	_ underline	7F	DEL	_ underline	65

†When these characters are copied from or to a tape, the characters remain the same and the code changes from or to ASCII to or from display code.

††These characters do not exist in display code. When the characters are copied from a tape, each ASCII character is changed to an alternate display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 61 (hexadecimal), from tape, it writes an uppercase A, 01 (octal).

†††A display code space always translates to an ASCII space.

§Character or code interpretation depends on context. Refer to Character Set Anomalies in the text.

TABLE A-6. EBCDIC 9-TRACK CODED TAPE CONVERSION

EBCDIC				6-Bit Display Code†††	
Code Conversion†		Character and Code Conversion††			
Code (Hex)	Character	Code (Hex)	Character	ASCII Character	Code (Octal)
40	space	00	NUL	space	55
4A	¢ cent sign	1C	IFS	[opening bracket	61
4B	. period	0E	SO	. period	57
4C	< less than	C0	{ opening brace	< less than	72
4D	(opening parenthesis	16	BS	(opening parenthesis	51
4E	+ plus	0B	VT	+ plus	45
4F	| vertical line	D0	} closing brace	! exclamation point	66
50	& ampersand	2E	ACK	& ampersand	67
5A	! exclamation point	01	SOH	] closing bracket	62
5B	$ dollar sign	37	EOT	$ dollar sign	53
5C	* asterisk	25	LF	* asterisk	47
5D	) closing parenthesis	05	HT	) closing parenthesis	52
5E	; semicolon	27	ESC	; semicolon	77
5F	¬ logical NOT	A1	~ tilde	^ caret	76
60	- hyphen (minus)	0D	CR	- hyphen (minus)	46
61	/ slant	0F	SI	/ slant	50
6B	, comma	0C	FF	, comma	56
6C	% percent sign§	2D	ENQ	% percent sign§	63§
6D	_ underline	07	DEL	_ underline	65
6E	> greater than	1E	IRS	> greater than	73
6F	? question mark	1F	IUS	? question mark	71
7A	: colon§	3F	SUB	: colon§	00§
7B	# number sign	03	ETX	# number sign	60
7C	@ commercial at	79	\ reverse slant	@ commercial at	74
7D	' apostrophe	2F	BEL	' apostrophe	70
7E	= equals	1D	IGS	= equals	54
7F	" quotation mark	02	STX	" quotation mark	64
C1	A	81	a	A	01
C2	B	82	b	B	02
C3	C	83	c	C	03
C4	D	84	d	D	04
C5	E	85	e	E	05
C6	F	86	f	F	06
C7	G	87	g	G	07
C8	H	88	h	H	10
C9	I	89	i	I	11
D1	J	91	j	J	12
D2	K	92	k	K	13
D3	L	93	l	L	14

TABLE A-6. EBCDIC 9-TRACK CODED TAPE CONVERSION (Contd)

EBCDIC				6-Bit Display Code†††	
Code Conversion†		Character and Code Conversion††			
Code (Hex)	Character	Code (Hex)	Character	ASCII Character	Code (Octal)
D4	M	94	m	M	15
D5	N	95	n	N	16
D6	O	96	o	O	17
D7	P	97	p	P	20
D8	Q	98	q	Q	21
D9	R	99	r	R	22
E0	\ reverse slant	6A	| vertical line	\ reverse slant	75
E2	S	A2	s	S	23
E3	T	A3	t	T	24
E4	U	A4	u	U	25
E5	V	A5	v	V	26
E6	W	A6	w	W	27
E7	X	A7	x	X	30
E8	Y	A8	y	Y	31
E9	Z	A9	z	Z	32
F0	0	10	DLE	0	33
F1	1	11	DC1	1	34
F2	2	12	DC2	2	35
F3	3	13	TM	3	36
F4	4	3C	DC4	4	37
F5	5	3D	NAK	5	40
F6	6	32	SYN	6	41
F7	7	26	ETB	7	42
F8	8	18	CAN	8	43
F9	9	19	EM	9	44

†When these characters are copied from or to a tape, the characters remain the same (except EBCDIC codes 4A (hexadecimal), 4F (hexadecimal), 5A (hexadecimal), and 5F (hexadecimal)) and the code changes from or to EBCDIC to or from display code.

††These characters do not exist in display code. When the characters are copied from a tape, each EBCDIC character is changed to an alternate display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 81 (hexadecimal), from tape, it writes an uppercase A, 01 (octal).

†††A display code space always translates to an EBCDIC space.

§Character or code interpretation depends on context. Refer to Character Set Anomalies in the text.

TABLE A-7. FULL EBCDIC CHARACTER SET

Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†	Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†	Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†
00	0000	NUL	4A	0112	¢ cent sign	A7	0247	x
01	0001	SOH	4B	0113	. period	A8	0250	y
02	0002	STX	4C	0114	< less than	A9	0251	z
03	0003	ETX	4D	0115	(open. paren.	AA	0252	undefined
04	0004	PF	4E	0116	+ plus	thru	thru	
05	0005	HT	4F	0117	| logical OR	BF	0277	undefined
06	0006	LC	50	0120	& ampersand	C0	0300	{ open. brace
07	0007	DEL	51	0121	undefined	C1	0301	A
08	0010	undefined	thru	thru		C2	0302	B
09	0011	undefined	59	0131	undefined	C3	0303	C
0A	0012	SMM	5A	0132	! exclam. point	C4	0304	D
0B	0013	VT	5B	0133	$ dollar sign	C5	0305	E
0C	0014	FF	5C	0134	* asterisk	C6	0306	F
0D	0015	CR	5D	0135	) clos. paren.	C7	0307	G
0E	0016	SO	5E	0136	; semicolon	C8	0310	H
0F	0017	SI	5F	0137	¬ logical NOT	C9	0311	I
10	0020	DLE	60	0140	- minus	CA	0312	undefined
11	0021	DC1	61	0141	/ slant	CB	0313	undefined
12	0022	DC2	62	0142	undefined	CC	0314	♪
13	0023	TM	thru	thru		CD	0315	undefined
14	0024	RES	69	0151	undefined	CE	0316	Ψ
15	0025	NL	6A	0152	¦ vertical line	CF	0317	undefined
16	0026	BS	6B	0153	, comma	D0	0320	} clos. brace
17	0027	IL	6C	0154	% percent sign	D1	0321	J
18	0030	CAN	6D	0155	_ underline	D2	0322	K
19	0031	EM	6E	0156	> greater than	D3	0323	L
1A	0032	CC	6F	0157	? question mark	D4	0324	M
1B	0033	CU1	70	0160	undefined	D5	0325	N
1C	0034	IFS	thru	thru		D6	0326	O
1D	0035	IGS	78	0170	undefined	D7	0327	P
1E	0036	IRS	79	0171	` grave accent	D8	0330	Q
1F	0037	IUS	7A	0172	: colon	D9	0331	R
20	0040	DS	7B	0173	# number sign	DA	0332	undefined
21	0041	SOS	7C	0174	@ commercial at	thru	thru	
22	0042	FS	7D	0175	' apostrophe	DF	0337	undefined
23	0043	undefined	7E	0176	= equals	E0	0340	\ reverse slant
24	0044	BYP	7F	0177	" quotation mark	E1	0341	undefined
25	0045	LF	80	0200	undefined	E2	0342	S
26	0046	ETBB	81	0201	a	E3	0343	T
27	0047	ESCE	82	0202	b	E4	0344	U

TABLE A-7. FULL EBCDIC CHARACTER SET (Contd)

Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†	Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†	Hexa-decimal EBCDIC Code	Octal 12-Bit EBCDIC Code	EBCDIC Graphic or Control Character†
28	0050	undefined	83	0203	c	E5	0345	V
29	0051	undefined	84	0204	d	E6	0346	W
2A	0052	SM	85	0205	e	E7	0347	X
2B	0053	CU2	86	0206	f	E8	0350	Y
2C	0054	undefined	87	0207	g	E9	0351	Z
2D	0055	ENQ	88	0210	h	EA	0352	undefined
2E	0056	ACK	89	0211	i	EB	0353	undefined
2F	0057	BEL	8A	0212	undefined	EC	0354	undefined
30	0060	undefined	thru	thru		ED	0355	undefined
31	0061	undefined	90	0220	undefined	thru	thru	
32	0062	SYN	91	0221	j	EF	0357	undefined
33	0063	undefined	92	0222	k	F0	0360	0
34	0064	PN	93	0223	l	F1	0361	1
35	0065	RS	94	0224	m	F2	0362	2
36	0066	UC	95	0225	n	F3	0363	3
37	0067	EOT	96	0226	o	F4	0364	4
38	0070	undefined	97	0227	p	F5	0365	5
39	0071	undefined	98	0230	q	F6	0366	6
3A	0072	undefined	99	0231	r	F7	0367	7
3B	0073	CU3	9A	0232	undefined	F8	0370	8
3C	0074	DC4	thru	thru		F9	0372	9
3D	0075	NAK	A0	0240	undefined	FA	0372	| vertical line
3E	0076	undefined	A1	0241	~ tilde	FB	0373	undefined
3F	0077	SUB	A2	0242	s	thru	thru	
40	0100	space	A3	0243	t	FF	0377	undefined
41	0101	undefined	A4	0244	u			
thru	thru		A5	0245	v			
49	0111	undefined	A6	0246	w			

†Graphic characters shown are those used on the IBM System/370 standard (PN) print train. Other devices support subsets or variations of this character graphic set.

TABLE A-8. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 9, 14, 16, 17, AND 18 (HASP, HPRE, 2780, 3270, AND 3780)

Terminal EBCDIC				Network ASCII (Normalized Mode Use)			
Hex. Code	Octal Code	Graphic†	Control Character††	Hex. Code†††	Octal Code†††	Graphic	Control Character††
00	000		NUL	00	000		null
01	001		SOH	01	001		start of header
02	002		STX	02	002		start of text
03	003		ETX	03	003		end of text
04	004		PF	20	040	space	
05	005		HT	09	011		horizontal tabulate
06	006		LC	20	040	space	
07	007		DEL	7F	177		delete
08	010		undefined	20	040	space	
09	011		undefined	20	040	space	
0A	012		SMM	20	040	space	
0B	013		VT	0B	013		vertical tabulate
0C	014		FF	0C	014		form feed
0D	015		CR	0D	015		carriage return
0E	016		SO	0E	016		shift out
0F	017		SI	0F	017		shift in
10	020		DLE	10	020		data link escape
11	021		DC1	11	021		device control 1
12	022		DC2	12	022		device control 2
13	023		TM	13	023		device control 3
14	024		RES	20	040	space	
15	025		NL	20	040	space	
16	026		BS	08	010		backspace
17	027		IL	20	040	space	
18	030		CAN	18	030		cancel
19	031		EM	19	031		end of medium
1A	032		CC	20	040	space	
1B	033		CU1	20	040	space	
1C	034		IFS	1C	034		file separator
1D	035		IGS	1D	035		group separator
1E	036		IRS	1E	036		record separator
1F	037		IUS	1F	037		unit separator
20	040		DS	20	040	space	
21	041		SOS	20	040	space	
22	042		FS	20	040	space	
23	043		undefined	20	040	space	
24	044		BYP	20	040	space	
25	045		LF	0A	012		linefeed
26	046		ETB or EOB	17	027		end of transmission block
27	047		ESC or PRE	1B	033		escape
28	050		undefined	20	040	space	
29	051		undefined	20	040	space	
2A	052		SM	20	040	space	
2B	053		CU2	20	040	space	
2C	054		undefined	20	040	space	
2D	055		ENQ	05	005		enquiry
2E	056		ACK	06	006		positive acknowledgment
2F	057		BEL	07	007		bell
30	060		undefined	20	040	space	
31	061		undefined	20	040	space	
32	062		SYN	16	026		synchronous idle
33	063		undefined	20	040	space	
34	064		PN	20	040	space	
35	065		RS	20	040	space	
36	066		UC	20	040	space	
37	067		EOT	04	004		end of transmission
38	070		undefined	20	040	space	
39	071		undefined	20	040	space	
3A	072		undefined	20	040	space	

TABLE A-8. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 9, 14, 16, 17, AND 18 (HASP, HPRE, 2780, 3270, AND 3780) (Contd)

Terminal EBCDIC				Network ASCII (Normalized Mode Use)			
Hex. Code	Octal Code	Graphic†	Control Character††	Hex. Code†††	Octal Code†††	Graphic	Control Character††
3B	073		CU3	20	040	space	
3C	074		DC4	14	024		device control 4
3D	075		NAK	15	025		negative acknowledgement
3E	076		undefined	20	040	space	
3F	077		SUB	1A	032		substitute
40	100	space		20	040	space	
41	101		undefined	20	040	space	
thru	thru						
49	111						
4A	112	¢		5B	133	[	
4B	113	.		2E	056	.	
4C	114	<		3C	074	<	
4D	115	(		28	050	(	
4E	116	+		2B	053	+	
4F	117	|		21	041	!	
50	120	&		26	046	&	
51	121		undefined	20	040	space	
thru	thru						
59	131						
5A	132	!		50	135	]	
5B	133	$		24	044	$	
5C	134	*		2A	052	*	
5D	135	)		29	051	)	
5E	136	;		3B	073	;	
5F	137	¬		5E	136	^	
60	140	-		2D	055	-	
61	141	/		2F	057	/	
62	142		undefined	20	040	space	
thru	thru						
69	151						
6A	152	¦		7C	174	¦	
6B	153	,		2C	054	,	
6C	154	%		25	045	%	
6D	155			5F	137		
6E	156	>		3E	076	>	
6F	157	?		3F	077	?	
70	160		undefined	20	040	space	
thru	thru						
78	170						
79	171	`		60	140	`	
7A	172	:		7A	172	:	
7B	173	#		23	043	#	
7C	174	@		40	100	@	
7D	175	'		27	047	'	
7E	176	=		3D	075	=	
7F	177	"		22	042	"	
80	200		undefined	20	040	space	
81	201	a		61	141	a	
82	202	b		62	142	b	
83	203	c		63	143	c	
84	204	d		64	144	d	
85	205	e		65	145	e	
86	206	f		66	146	f	
87	207	g		67	147	g	
88	210	h		68	150	h	
89	211	i		69	151	i	
8A	212		undefined	20	040	space	
thru	thru						
90	220						

TABLE A-8. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 9, 14, 16, 17, AND 18 (HASP, HPRE, 2780, 3270, AND 3780) (Contd)

Terminal EBCDIC				Network ASCII (Normalized Mode Use)			
Hex. Code	Octal Code	Graphic†	Control Character††	Hex. Code†††	Octal Code†††	Graphic	Control Character††
91	221	j		6A	152	j	
92	222	k		6B	153	k	
93	223	l		6C	154	l	
94	224	m		6D	155	m	
95	225	n		6E	156	n	
96	226	o		6F	157	o	
97	227	p		70	160	p	
98	230	q		71	161	q	
99	231	r		72	162	r	
9A thru A0	232 thru 240		undefined	20	040	space	
A1	241	~		7E	176	~	
A2	242	s		73	163	s	
A3	243	t		74	164	t	
A4	244	u		75	165	u	
A5	245	v		76	166	v	
A6	246	w		77	167	w	
A7	247	x		78	170	x	
A8	250	y		79	171	y	
A9	251	z		7A	172	z	
AA thru BF	252 thru 277		undefined	20	040	space	
C0	300	{		7B	173	{	
C1	301	A		41	101	A	
C2	302	B		42	102	B	
C3	303	C		43	103	C	
C4	304	D		44	104	D	
C5	305	E		45	105	E	
C6	306	F		46	106	F	
C7	307	G		47	107	G	
C8	310	H		48	110	H	
C9	311	I		49	111	I	
CA	312		undefined	20	040	space	
CB	313		undefined	20	040	space	
CC	314	♪		20	040	space	
CD	315		undefined	20	040	space	
CE	316	♆		20	040	space	
CF	317		undefined	20	040	space	
D0	320	}		7E	175	}	
D1	321	J		4A	112	J	
D2	322	K		4B	113	K	
D3	323	L		4C	114	L	
D4	324	M		4D	115	M	
D5	325	N		4E	116	N	
D6	326	O		4F	117	O	
D7	327	P		50	120	P	
D8	330	Q		51	121	Q	
D9	331	R		52	122	R	
DA thru DF	332 thru 337		undefined	20	040	space	
E0	340	\		5C	134	\	
E1	341		undefined	20	040	space	
E2	342	S		53	123	S	
E3	343	T		54	124	T	
E4	344	U		55	125	U	
E5	345	V		56	126	V	

TABLE A-8. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 9, 14, 16, 17, AND 18 (HASP, HPRE, 2780, 3270, AND 3780) (Contd)

Terminal EBCDIC				Network ASCII (Normalized Mode Use)			
Hex. Code	Octal Code	Graphic†	Control Character††	Hex. Code†††	Octal Code†††	Graphic	Control Character††
E6	346	W		57	127	W	
E7	347	X		58	130	X	
E8	350	Y		59	131	Y	
E9	351	Z		5A	132	Z	
EA	352		undefined	20	040	space	
EB	353		undefined	20	040	space	
EC	354	┤		20	040	space	
ED	355		undefined	20	040	space	
thru	thru						
EF	357						
F0	360	0		30	060	0	
F1	361	1		31	061	1	
F2	362	2		32	062	2	
F3	363	3		33	063	3	
F4	364	4		34	064	4	
F5	365	5		35	065	5	
F6	366	6		36	066	6	
F7	367	7		37	067	7	
F8	370	8		38	070	8	
F9	371	9		39	071	9	
FA	372	|		20	040	space	
FB	373		undefined	20	040	space	
thru	thru						
FF	377						

†Graphic characters shown are those used on the IBM System/370 standard (PN) print train. Other devices support subsets or variations of this character graphic set.

††Not used for output to line printers. Translation to a space (100 octal) occurs.

†††Shown with zero parity (eighth or uppermost bit is always zero).

TABLE A-9. EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC) WITH PUNCHED CARD CODES AND 6-BIT DISPLAY CODE TRANSLATION (NETWORK TRANSPARENT CARD READERS, TERMINAL CLASSES 9, 14, 16, AND 17; HASP, HPRE, 2780 AND 3780)

EBCDIC BITS	1ST HEX 2ND	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL 12-0-1-8-9 space 2D	DLE 12-11-1-8-9 space 2D	DS 11-0-1-8-9 space 2D	12-11-0-1-8-9 space 2D	space no punch space 2D	& 12 (0-7-8) ^ 37	- 11 - 26	12-11-0 space 2D	12-0-1-8 space 2D	12-11-1-8 space 2D	11-0-1-8 space 2D	12-11-0-1-8 space 2D	12-0 space 2D	11-0 space 2D	\ 0-2-8 (12-5-8) ≥ 3D	0 0 0 1B
0001	1	SOH 12-1-9 space 2D	DC1 11-1-9 space 2D	SOS 0-1-9 space 2D	1-9 space 2D	12-0-1-9 space 2D	12-11-1-9 space 2D	/ 0-1 / 28	12-11-0-1-9 space 2D	a 12-0-1 A 01	j 12-11-1 J 0A	11-0-1 space 2D	12-11-0-1 space 2D	A 12-1 A 01	J 11-1 J 0A	11-0-1-9 space 2D	1 1 1 1C
0010	2	STX 12-2-9 space 2D	DC2 11-2-9 space 2D	FS 0-2-9 (4-8) space 2D	SYN 2-9 space 2D	12-0-2-9 space 2D	12-11-2-9 space 2D	11-0-2-9 space 2D	12-11-0-2-9 space 2D	b 12-0-2 B 02	k 12-11-2 K 0B	s 11-0-2 S 13	12-11-0-2 space 2D	B 12-2 B 02	K 11-2 K 0B	S 0-2 S 13	2 2 2 1D
0011	3	ETX 12-3-9 space 2D	TM 11-3-9 space 2D	0-3-9 space 2D	3-9 space 2D	12-0-3-9 space 2D	12-11-3-9 space 2D	11-0-3-9 space 2D	12-11-0-3-9 space 2D	c 12-0-3 C 03	l 12-11-3 L 0C	t 11-0-3 T 14	12-11-0-3 space 2D	C 12-3 C 03	L 11-3 L 0C	T 0-3 T 14	3 3 3 1E
0100	4	PF 12-4-9 space 2D	RES 11-4-9 space 2D	BYP 0-4-9 space 2D	PN 4-9 space 2D	12-0-4-9 space 2D	12-11-4-9 space 2D	11-0-4-9 space 2D	12-11-0-4-9 space 2D	d 12-0-4 D 04	m 12-11-4 M 0D	u 11-0-4 U 15	12-11-0-4 space 2D	D 12-4 D 04	M 11-4 M 0D	U 0-4 U 15	4 4 4 1F
0101	5	HT 12-5-9 space 2D	NL 11-5-9 space 2D	LF 0-5-9 space 2D	RS 5-9 space 2D	12-0-5-9 space 2D	12-11-5-9 space 2D	11-0-5-9 space 2D	12-11-0-5-9 space 2D	e 12-0-5 E 05	n 12-11-5 N 0E	v 11-0-5 V 16	12-11-0-5 space 2D	E 12-5 E 05	N 11-5 N 0E	V 0-5 V 16	5 5 5 20
0110	6	LC 12-6-9 space 2D	BS 11-6-9 space 2D	ETB 0-6-9 space 2D	UC 6-9 space 2D	12-0-6-9 space 2D	12-11-6-9 space 2D	11-0-6-9 space 2D	12-11-0-6-9 space 2D	f 12-0-6 F 06	o 12-11-6 O 0F	w 11-0-6 W 17	12-11-0-6 space 2D	F 12-6 F 06	O 11-6 O 0F	W 0-6 W 17	6 6 6 21
0111	7	DEL 12-7-9 space 2D	IL 11-7-9 space 2D	ESC 0-7-9 space 2D	EOT 7-9 space 2D	12-0-7-9 space 2D	12-11-7-9 space 2D	11-0-7-9 space 2D	12-11-0-7-9 space 2D	g 12-0-7 G 07	p 12-11-7 P 10	x 11-0-7 X 18	12-11-0-7 space 2D	G 12-7 G 07	P 11-7 P 10	X 0-7 X 18	7 7 7 22
1000	8	GE 12-8-9 space 2D	CAN 11-8-9 space 2D	0-8-9 space 2D	8-9 space 2D	12-0-8-9 space 2D	12-11-8-9 space 2D	11-0-8-9 space 2D	12-11-0-8-9 space 2D	h 12-0-8 H 08	q 12-11-8 Q 11	y 11-0-8 Y 19	12-11-0-8 space 2D	H 12-8 H 08	Q 11-8 Q 11	Y 0-8 Y 19	8 8 8 23
1001	9	RLF 12-1-8-9 space 2D	EM 11-1-8-9 space 2D	0-1-8-9 space 2D	1-8-9 space 2D	12-1-8 space 2D	11-1-8 space 2D	0-1-8 space 2D	1-8 space 2D	i 12-0-9 I 09	r 12-11-9 R 12	z 11-0-9 Z 1A	12-11-0-9 space 2D	I 12-9 I 09	R 11-9 R 12	Z 0-9 Z 1A	9 9 9 24
1010	A	SMM 12-2-8-9 space 2D	CC 11-2-8-9 space 2D	SM 0-2-8-9 space 2D	2-8-9 space 2D	¢ 12-2-8 (7-8) [31	! 11-2-8 (0-2-8)] 32	, 12-11 space 2D	. 2-8 : 00	12-0-2-8 space 2D	12-11-2-8 space 2D	11-0-2-8 space 2D	12-11-0-2-8 space 2D	12-0-2-8-9 space 2D	12-11-2-8-9 space 2D	11-0-2-8-9 space 2D	' (LVM) 12-11-0-2-8-9 space 2D
1011	B	VT 12-3-8-9 space 2D	CU1 11-3-8-9 space 2D	CU2 0-3-8-9 space 2D	CU3 3-8-9 space 2D	. 12-3-8 . 2F	$ 11-3-8 $ 2B	, 0-3-8 , 2E	# 3-8 (0-6-8) ≡ 30	12-0-3-8 space 2D	12-11-3-8 space 2D	11-0-3-8 space 2D	12-11-0-3-8 space 2D	12-0-3-8-9 space 2D	12-11-3-8-9 space 2D	11-0-3-8-9 space 2D	12-11-0-3-8-9 space 2D
1100	C	FF 12-4-8-9 space 2D	IFS 11-4-8-9 space 2D	0-4-8-9 space 2D	DC4 4-8-9 space 2D	< 12-4-8 (12-0) < 3A	* 11-4-8 (12-5-8) * 27	% 0-4-8 (6-8) % 33	@ 4-8 (5-8) ≤ 3C	12-0-4-8 space 2D	12-11-4-8 space 2D	11-0-4-8 space 2D	12-11-0-4-8 space 2D	12-0-4-8-9 space 2D	12-11-4-8-9 space 2D	11-0-4-8-9 space 2D	12-11-0-4-8-9 space 2D
1101	D	CR 12-5-8-9 space 2D	IGS 11-5-8-9 space 2D	ENQ 0-5-8-9 space 2D	NAK 5-8-9 space 2D	(12-5-8 (0-4-8) (29	) 11-5-8 (12-4-8)) 2A	⌐ 0-5-8 ⌐ 35	' 5-8 (11-5-8) ↑ 38	12-0-5-8 space 2D	12-11-5-8 space 2D	11-0-5-8 space 2D	12-11-0-5-8 space 2D	12-0-5-8-9 space 2D	12-11-5-8-9 space 2D	11-0-5-8-9 space 2D	12-11-0-5-8-9 space 2D
1110	E	SO 12-6-8-9 space 2D	IRS 11-6-8-9 space 2D	ACK 0-6-8-9 space 2D	6-8-9 space 2D	+ 12-6-8 (12) + 25	, 11-6-8 (12-7-8) ; 3F	> 0-6-8 (11-7-8) > 3B	= 6-8 (3-8) = 2C	12-0-6-8 space 2D	12-11-6-8 space 2D	11-0-6-8 space 2D	12-11-0-6-8 space 2D	12-0-6-8-9 space 2D	12-11-6-8-9 space 2D	11-0-6-8-9 space 2D	12-11-0-6-8-9 space 2D
1111	F	SI 12-7-8-9 space 2D	IUS 11-7-8-9 space 2D	BEL 0-7-8-9 space 2D	SUB 7-8-9 space 2D	' 12-7-8 (11-0) ↘ 36	¬ 11-7-8 (12-6-8) ¬ 3E	? 0-7-8 (11-6-8) ↓ 39	" 7-8 (4-8) ≠ 34	12-0-7-8 space 2D	12-11-7-8 space 2D	11-0-7-8 space 2D	12-11-0-7-8 space 2D	12-0-7-8-9 space 2D	12-11-7-8-9 space 2D	11-0-7-8-9 space 2D	EO 12-11-0-7-8-9 space 2D

LEGEND: EBCDIC Character → + / 12-6-8 (029 Card Code) / (12) (026 Card Code) / CDC Graphic Character → + 25 (Display Code (Hexadecimal))

TABLE A-10. 6-BIT DISPLAY CODE WITH EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC) AND PUNCHED CODE TRANSLATIONS (LINE PRINTERS AND CARD PUNCHES, TERMINAL CLASSES 9, 14, 16, 17, AND 18; HASP, HPRE, 2780, 3270, AND 3780)

DISPLAY CODE BITS	1ST HEX 2ND	00 0	01 1	10 2	11 3	DISPLAY CODE BITS	1ST HEX 2ND	00 0	01 1	10 2	11 3
0000	0	: 2-8 : 7A	P 11-7 P D7	5 5 5 F5	≡ 3-8 (0-6-8) # 7B	1000	8	H 12-8 H C8	X 0-7 X E7	/ 0-1 / 61	↑ 5-8 (11-5-8) , 7D
0001	1	A 12-1 A C1	q 11-8 Q D8	6 6 6 F6	[12-2-8 (7-8) ¢ 4A	1001	9	I 12-9 I C9	Y 0-8 Y E8	(12-5-8 (0-4-8) (4D	↓ 0-7-8 (11-6-8) ? 6F
0010	2	B 12-2 B C2	R 11-9 R D9	7 7 7 F7	] 11-2-8 (0-2-8) ! 5A	1010	A	J 11-1 J D1	Z 0-9 Z E9	) 11-5-8 (12-4-8)) 5D	< 12-4-8 (12-0) < 4C
0011	3	C 12-3 C C3	S 0-2 S E2	8 8 8 F8	% 0-4-8 (6-8) % 6C	1011	B	K 11-2 K D2	0 0 0 F0	$ 11-3-8 $ 5B	> 0-6-8 (11-7-8) > 6E
0100	4	D 12-4 D C4	T 0-3 T E3	9 9 9 F9	≠ 7-8 (4-8) " 7F	1100	C	L 11-3 L D3	1 1 1 F1	= 6-8 (3-8) = 7E	≤ 4-8 (5-8) @ 7C
0101	5	E 12-5 E C5	U 0-4 U E4	+ 12-6-8 (12) + 4E	⌐ 0-5-8 _ 6D	1101	D	M 11-4 M D4	2 2 2 F2	SP no punch 40	≥ 0-2-8 (12-5-8) \ E0
0110	6	F 12-6 F C6	V 0-5 V E5	- 11 - 60	∨ 12-7-8 (11-0) ! 4F	1110	E	N 11-5 N D5	3 3 3 F3	, 0-3-8 , 6B	¬ 11-7-8 (12-6-8) ¬ 5F
0111	7	G 12-7 G C7	W 0-6 W E6	* 11-4-8 * 5C	∧ 12 (0-7-8) & 50	1111	F	O 11-6 O D6	4 4 4 F4	. 12-3-8 . 4B	; 11-6-8 (12-7-8) ; 5E

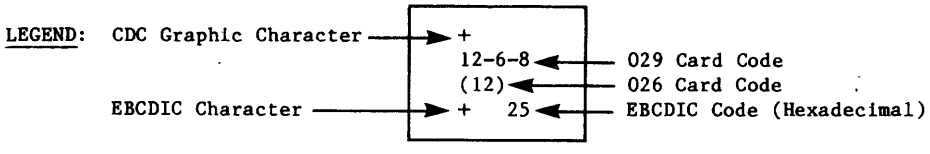


TABLE A-11. EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC) WITH PUNCHED CARD CODES (NETWORK TRANSPARENT MODE CARD READERS, TERMINAL CLASSES 9, 14, 16, 17, HASP, HPRE, 2780, AND 3780)

Bits b8 b7 b6 b5		0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
b4 b3 b2 b1	ROW \ COL	0	1	2	3	4	5	6	7	8	9	10 (A)	11 (B)	12 (C)	13 (D)	14 (E)	15 (F)
0 0 0 0	0	12-0-1-8-9 NUL 00	12-11-1-8-9 DLE 10	no punch SP 40	0 0 F0	4-8 @ 7C	11-7 P D7	1-8 ` 79	12-11-7 p 97	11-0-1-8-9 DS 20	12-11-0-1-8-9 30	12-0-1-9 41	12-11-8-9 58	12-11-0-6-9 F6 76	12-11-7-8 9F	12-11-0-8 B8	12-11-4-8-9 DC
0 0 0 1	1	12-1-9 SOH 01	11-1-9 DC1 11	12-7-8 | 4F	1 1 F1	12-1 A C1	11-8 Q D8	12-0-1 a 81	12-11-8 q 98	0-1-9 SOS 21	1-9 31	12-0-2-9 42	11-1-8 59	12-11-0-7-9 F7 77	11-0-1-8 A0	12-11-0-9 B9	12-11-5-8-9 DD
0 0 1 0	2	12-2-9 STX 02	11-2-9 DC2 12	7-8 " 7F	2 2 F2	12-2 B C2	11-9 R D9	12-0-2 b 82	12-11-9 r 99	0-2-9 FS 22	11-2-8-9 CC 1A	12-0-3-9 43	11-0-2-9 62	12-11-0-8-9 F8 78	11-0-2-8 AA	12-11-0-2-8 BA	12-11-6-8-9 DE
0 0 1 1	3	12-3-9 ETX 03	11-3-9 TM 13	3-8 # 7B	3 3 F3	12-3 C C3	0-2 S E2	12-0-3 c 83	11-0-2 s A2	0-3-9 23	3-9 33	12-0-4-9 44	11-0-3-9 63	12-0-1-8 80	11-0-3-8 AB	12-11-0-3-8 BB	12-11-7-8-9 DF
0 1 0 0	4	7-9 EOT 37	4-8-9 DC4 3C	11-3-8 $ 5B	4 4 F4	12-4 D C4	0-3 T E3	12-0-4 d 84	11-0-3 t A3	0-4-9 BYP 24	4-9 PN 34	12-0-5-9 45	11-0-4-9 64	12-0-2-8 8A	11-0-4-8 AC	12-11-0-4-8 BC	11-0-2-8-9 EA
0 1 0 1	5	0-5-8-9 ENQ 2D	5-8-9 NAK 3D	0-4-8 % 6C	5 5 F5	12-5 E C5	0-4 U E4	12-0-5 e 85	11-0-4 u A4	11-5-9 NL 15	5-9 RS 35	12-0-6-9 46	11-0-5-9 65	12-0-3-8 8B	11-0-5-8 AD	12-11-0-5-8 BD	11-0-3-8-9 EB
0 1 1 0	6	0-6-8-9 ACK 2E	2-9 SYN 32	12 & 50	6 6 F6	12-6 F C6	0-5 V E5	12-0-6 f 86	11-0-5 v A5	12-6-9 LC 06	6-9 UC 36	12-0-7-9 47	11-0-6-9 66	12-0-4-8 8C	11-0-6-8 AE	12-11-0-6-8 BE	11-0-4-8-9 ¶ EC
0 1 1 1	7	0-7-8-9 BEL 2F	0-6-9 ETB 26	5-8 ' 7D	7 7 F7	12-7 G C7	0-6 W E6	12-0-7 g 87	11-0-6 w A6	11-7-9 IL 17	12-8-9 GE 08	12-0-8-9 48	11-0-7-9 67	12-0-5-8 8D	11-0-7-8 AF	12-11-0-7-8 BF	11-0-5-8-9 ED
1 0 0 0	8	11-6-9 BS 16	11-8-9 CAN 18	12-5-8 (4D	8 8 F8	12-8 H C8	0-7 X E7	12-0-8 h 88	11-0-7 x A7	0-8-9 28	8-9 38	12-1-8 49	11-0-8-9 68	12-0-6-8 8E	12-11-0-1-8 B0	12-0-2-8-9 CA	11-0-6-8-9 EE
1 0 0 1	9	12-5-9 HT 05	11-1-8-9 EM 19	11-5-8) 5D	9 9 F9	12-9 I C9	0-8 Y E8	12-0-9 i 89	11-0-8 y A8	0-1-8-9 29	1-8-9 39	12-11-1-9 51	0-1-8 69	12-0-7-8 8F	12-11-0-1 B1	12-0-3-8-9 CB	11-0-7-8-9 EF
1 0 1 0	10 (A)	0-5-9 LF 25	7-8-9 SUB 3F	11-4-8 * 5C	2-8 : 7A	11-1 J D1	0-9 Z E9	12-11-1 j 91	11-0-9 z A9	0-2-8-9 SM 2A	2-8-9 3A	12-11-2-9 52	12-11-0 70	12-11-1-8 90	12-11-0-2 B2	12-0-4-8-9 ♪ CC	12-11-0-2-8-9 | (LVM) FA
1 0 1 1	11 (B)	12-3-8-9 VT 0B	0-7-9 ESC 27	12-6-8 + 4E	11-6-8 ; 5E	11-2 K D2	12-2-8 ¢ 4A	12-11-2 k 92	12-0 { C0	0-3-8-9 CU2 2B	3-8-9 CU3 3B	12-11-3-9 53	12-11-0-1-9 F1 71	12-11-2-8 9A	12-11-0-3 B3	12-0-5-8-9 CD	12-11-0-3-8-9 FB
1 1 0 0	12 (C)	12-4-8-9 FF 0C	11-4-8-9 IFS 1C	0-3-8 , 6B	12-4-8 < 4C	11-3 L D3	0-2-8 \ E0	12-11-3 l 93	12-11 ¦ 6A	0-4-8-9 2C	12-4-9 PF 04	12-11-4-9 54	12-11-0-2-9 F2 72	12-11-3-8 9B	12-11-0-4 B4	12-0-6-8-9 Ψ CE	12-11-0-4-8-9 FC
1 1 0 1	13 (D)	12-5-8-9 CR 0D	11-5-8-9 IGS 1D	11 - 60	6-8 = 7E	11-4 M D4	11-2-8 ! 5A	12-11-4 m 94	11-0 } D0	12-1-8-9 RLF 09	11-4-9 RES 14	12-11-5-9 55	12-11-0-3-9 F3 73	12-11-4-8 9C	12-11-0-5 B5	12-0-7-8-9 CF	12-11-0-5-8-9 FD
1 1 1 0	14 (E)	12-6-8-9 SO 0E	11-6-8-9 IRS 1E	12-3-8 . 4B	0-6-8 > 6E	11-5 N D5	11-7-8 ¬ 5F	12-11-5 n 95	11-0-1 ~ A1	12-2-8-9 SMM 0A	6-8-9 3E	12-11-6-9 56	12-11-0-4-9 F4 74	12-11-5-8 9D	12-11-0-6 B6	12-11-2-8-9 DA	12-11-0-6-8-9 FE
1 1 1 1	15 (F)	12-7-8-9 SI 0F	11-7-8-9 IUS 1F	0-1 / 61	0-7-8 ? 6F	11-6 O D6	0-5-8 _ 6D	12-11-6 o 96	12-7-9 DEL 07	11-3-8-9 CU1 1B	11-0-1-9 E1	12-11-7-9 57	12-11-0-5-9 F5 75	12-11-6-8 9E	12-11-0-7 B7	12-11-3-8-9 DB	12-11-0-7-8-9 FF

LEGEND:

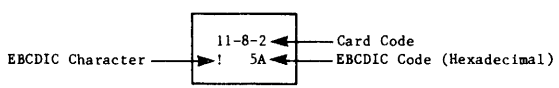


TABLE A-12. CHARACTER CODE TRANSLATIONS, ASCII CHARACTER SET CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code‡	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
00	000		NUL or (@)	00	000		null
03	003	▲	ETX or (C)	03	003		end of text
05	005		ENQ or WRU or (E)	05	005		enquiry
06	006		ACK or RU or (F)	06	006		positive acknowledgement
09	011		HT or (I)	09	011		horizontal tabulate
0A	012		LF or NL or ↓ or (J)	0A	012		linefeed
0C	014		FF or FORM or (L)	0C	014		formfeed
0F	017	≫	SI or (O)	0F	017		shift in
11	021		DC1 or X-ON or (Q)	11	021		device control 1
12	022		DC2 or TAPE or (R)	12	022		device control 2
14	024		DC4 or TAPE or (T)	14	024		device control 4
17	027		ETB or (W)	17	027		end transmission block
18	030		CAN or CLEAR or (X)	18	030		cancel
1B	033		ESC or ESCAPE or ([)	1B	033		escape
1D	035		GS or (])	1D	035		group separator
1E	036		RS or (^)	1E	036		record separator
21	041	!		21	041	!	
22	042	"		22	042	"	
24	044	$		24	044	$	
27	047	'		27	047	'	
28	050	(		28	050	(	
2B	053	+		2B	053	+	
2D	055	-		2D	055	-	
2E	056	.		2E	056	.	
30	060	0		30	060	0	
33	063	3		33	063	3	
35	065	5		35	065	5	
36	066	6		36	066	6	
39	071	9		39	071	9	
3A	072	:		3A	072	:	
3C	074	<		3C	074	<	
3F	077	?		3F	077	?	
41	101	A		41	101	A	
42	102	B		42	102	B	
44	104	D		44	104	D	
47	107	G		47	107	G	
48	110	H		48	110	H	
4B	113	K		4B	113	K	
4D	115	M		4D	115	M	
4E	116	N		4E	116	N	
50	120	P		50	120	P	
53	123	S		53	123	S	
55	125	U		55	125	U	
56	126	V		56	126	V	
59	131	Y		59	131	Y	
5A	132	Z		5A	132	Z	
5C	134	\		5C	134	\	
5F	137	_ or ←		5F	137		
60	140	‾		60	140	‾	
63	143	c		63	143	c	
65	145	e		65	145	e	
66	146	f		66	146	f	
69	151	i		69	151	i	
6A	152	j		6A	152	j	
6C	154	l		6C	154	l	
6F	157	o		6F	157	o	
71	161	q		71	161	q	
72	162	r		72	162	r	

TABLE A-12. CHARACTER CODE TRANSLATIONS, ASCII CHARACTER SET CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
74	164	t		74	164	t	
77	167	w		77	167	w	
78	170	x		78	170	x	
7B	173	{		7B	173	{	
7C	174	¦ or ↑ or |		7C	174	¦	
7D	175	}		7D	175	}	
7E	176	~ or ¬		7E	176	~	
81	201		SOH or (A)	01	001		start of header
82	202		STX or (B)	02	002		start of text
84	204		EOT or (D)	04	004		end of transmission
87	207		BELL or (G)	07	007		bell
88	210		BS or ← or (H)	08	010		backspace
8B	213		VT or (K)	0B	013		vertical tabulate
8D	215		CR or RETURN or (M)	0D	015		carriage return
8E	216	≪	SO or (N)	0E	016		shift out
90	220		DLE or (P)	10	020		data link escape
93	223		DC3 or X-OFF or (S)	13	023		device control 3
95	225		NAK or → or (U)	15	025		negative acknowledgement
96	226		SYN or LINE CLEAR or (V)	16	026		synchronous idle
99	231		EM or RESET or (Y)	19	031		end of medium
9A	232		SUB or ↑ or (Z)	1A	032		substitute
9C	234		FS or (\)	1C	034		file separator
9F	237		US or (−)	1F	037		unit separator
A0	240	SPACE or blank		20	040	space	
A3	243	#		23	043	#	
A5	245	%		25	045	%	
A6	246	&		26	046	&	
A9	251	)		29	051	)	
AA	252	*		2A	052	*	
AC	254	,		2C	054	,	
AF	257	/		2F	057	/	
B1	261	1		31	061	1	
B2	262	2		32	062	2	
B4	264	4		34	064	4	
B7	267	7		37	067	7	
B8	270	8		38	070	8	
BB	273	;		3B	073	;	
BD	275	=		3D	075	=	
BE	276	>		3E	076	>	
C0	300	@		40	100	@	
C3	303	C		43	103	C	
C5	305	E		45	105	E	
C6	306	F		46	106	F	
C9	311	I		49	111	I	
CA	312	J		4A	112	J	
CC	314	L		4C	114	L	
CF	317	O		4F	117	O	
D1	321	Q		51	121	Q	
D2	322	R		52	122	R	
D4	324	T		54	124	T	
D7	327	W		57	127	W	
D8	330	X		58	130	X	
DB	333	[		5B	133	[	
DD	335	]		5D	135	]	
DE	336	^ or ¬		5E	136	^	
E1	341	a		61	141	a	

TABLE A-12. CHARACTER CODE TRANSLATIONS, ASCII CHARACTER SET CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
E2	342	b		62	142	b	
E4	344	d		64	144	d	
E7	347	g		67	147	g	
E8	350	h		68	150	h	
EB	353	k		6B	153	k	
ED	355	m		6D	155	m	
EE	356	n		6E	156	n	
F0	360	p		70	160	p	
F3	363	s		73	163	s	
F5	365	u		75	165	u	
F6	366	v		76	166	v	
F9	371	y		79	171	y	
FA	372	z		7A	172	z	
FF	377	■	DEL or RUBOUT	7F	177		delete

†Shown with even parity, which is the default for these terminal classes (unless PA=N or PA=I, an application program receives the same code as in normalized mode).

††A circle around a character indicates that the character key is pressed in conjunction with a CTL, CTRL, CNTRL, or CONTROL key to generate the code.

†††Shown with zero parity (eighth or uppermost bit is always zero).

TABLE A-13. CHARACTER CODE TRANSLATIONS, APL TYPEWRITER-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
00	000		NUL or (@)	00	000		null
03	003	▲	ETX or (C)	03	003		end of text
05	005		ENQ or WRU or (E)	05	005		enquiry
06	006		ACK or RU or (F)	06	006		positive acknowledgement
09	011		HT or (I)	09	011		horizontal tabulate
0A	012		LF or NL or ↓ or (J)	0A	012		linefeed
0C	014		FF or FORM or (L)	0C	014		formfeed
0F	017	⩾	SI or (O)	0F	017		shift in
11	021		DC1 or X-ON or (Q)	11	021		device control 1
12	022		DC2 or TAPE or (R)	12	022		device control 2
14	024		DC4 or TAPE or (T)	14	024		device control 4
17	027		ETB or (W)	17	027		end transmission block
18	030		CAN or CLEAR or (X)	18	030		cancel
1B	033		ESC or ESCAPE or ([)	1B	033		escape
1D	035		GS or (])	1D	035		group separator
1E	036		RS or (^)	1E	036		record separator
21	041	.		23	043	.	
22	042	;		29	052	;	
24	044	<		40	100	<	
27	047	⍦		5D	135	⍦	
28	050	∨		21	041	∨	
2B	053	+		25	045	+	
2D	055	÷		2B	053	÷	
2E	056	.		2E	056	.	
30	060	0		30	060	0	
33	063	3		33	063	3	
35	065	5		35	065	5	
36	066	6		36	066	6	
39	071	9		39	071	9	
3A	072	(		28	050	(	
3C	074	;		3B	073	;	
3F	077	\		5C	134	\	
41	101	⍺		61	141	⍺	
42	102	⊥		62	142	⊥	
44	104	⌊		64	144	⌊	
47	107	∇		67	147	∇	
48	110	Δ		68	150	Δ	
4B	113	`		27	047	`	
4D	115	|		6D	155	|	
4E	116	⍑		6E	156	⍑	
50	120	*		2A	052	*	
53	123	⌈		73	163	⌈	
55	125	↓		75	165	↓	
56	126	∪		76	166	∪	
59	131	↑		79	171	↑	
5A	132	⊂		7A	172	⊂	
5C	134	⊢		7E	176	⊢	
5F	137	-		2D	055	-	
60	140	◇		60	140	◇	
63	143	C		43	103	C	
65	145	E		45	105	E	
66	146	F		46	106	F	
69	151	I		47	111	I	
6A	152	J		4A	112	J	
6C	154	L		4C	114	L	
6F	157	O		4F	117	O	
71	161	Q		51	121	Q	
72	162	R		52	122	R	

TABLE A-13. CHARACTER CODE TRANSLATIONS, APL TYPEWRITER-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
74	164	T		54	124	T	
77	167	W		57	127	W	
78	170	X		58	130	X	
7B	173	{		7B	173	{	
7C	174	⊤		6B	153	⊤	
7D	175	}		7D	175	}	
7E	176	$		24	044	$	
81	201		SOH or (A)	01	001		start of header
82	202		STX or (B)	02	002		start of text
84	204		EOT or (D)	04	004		end of transmission
87	207		BELL or (G)	07	007		bell
88	210		BS or ← or (H)	08	010		backspace
8B	213		VT or (K)	0B	013		vertical tabulate
8D	215		CR or RETURN or (M)	0D	015		carriage return
8E	216	⋘	SO or (N)	0E	016	⋘	shift out
90	220		DLE or (P)	10	020		data link escape
93	223		DC3 or X-OFF or (S)	13	023		device control 3
95	225		NAK or → or (U)	15	025		negative acknowledgement
96	226		SYN or LINE CLEAR or (V)	16	026		synchronous idle
99	231		EM or RESET or (Y)	19	031		end of medium
9A	232		SUB or ↑ or (Z)	1A	032		substitute
9C	234		FS or (\)	1C	034		file separator
9F	237		US or (−)	1F	037		unit separator
A0	240	SPACE or blank		20	040	space	
A3	243	<		3C	074	<	
A5	245	=		3D	075	=	
A6	246	>		3E	076	>	
A9	251	∧		26	046	∧	
AA	252	≠		22	042	≠	
AC	254	,		2C	054	,	
AF	257	/		2F	057	/	
B1	261	1		31	061	1	
B2	262	2		32	062	2	
B4	264	4		34	064	4	
B7	267	7		37	067	7	
B8	270	8		38	070	8	
BB	273	[		5B	133	[	
BD	275	⍉		66	146	⍉	
BE	276	:		3A	072	:	
C0	300	¯		5E	136	¯	
C3	303	∩		63	143	∩	
C5	305	∈		65	145	∈	
C6	306			5F	137		
C9	311	⍨		69	151	⍨	
CA	312	∘		6A	152	∘	
CC	314	□		6C	154	□	
CF	317	○		6F	157	○	
D1	321	?		3F	077	?	
D2	322	⍴		72	162	⍴	
D4	324	~		74	164	~	
D7	327	⍵		77	167	⍵	
D8	330	∪		78	170	∪	
DB	333	←		70	160	←	
DD	335	→		71	161	→	
DE	336	⍋		7C	174	⍋	
E1	341	A		41	101	A	

TABLE A-13. CHARACTER CODE TRANSLATIONS, APL TYPEWRITER-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
E2	342	B		42	102	B	
E4	344	D		44	104	D	
E7	347	G		47	107	G	
E8	350	H		48	110	H	
EB	353	K		4B	113	K	
ED	355	M		4D	115	M	
EE	356	N		4E	116	N	
F0	360	P		50	120	P	
F3	363	S		53	123	S	
F5	365	U		55	125	U	
F6	366	V		56	126	V	
F9	371	Y		59	131	Y	
FA	372	Z		5A	132	Z	
FF	377	■	DEL or RUBOUT	7F	177		delete

†Shown with even parity, which is the default for these terminal classes (unless PA=N or PA=I, an application program receives the same code as in normalized mode).

††A circle around a character indicates that the character key is pressed in conjunction with a CTL, CTRL, CNTRL, or CONTROL key to generate the code.

†††Shown with zero parity (eighth or uppermost bit is always zero).

TABLE A-14. CHARACTER CODE TRANSLATIONS, APL BIT-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, AND M40)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
00	000		NUL or (@)	00	000		null
03	003	▲	ETX or (C)	03	003		end of text
05	005		ENQ or WRU or (E)	05	005		enquiry
06	006		ACK or RU or (F)	06	006		positive acknowledgement
09	011		HT or (I)	09	011		horizontal tabulate
0A	012		LF or NL or ↓ or (J)	0A	012		linefeed
0C	014		FF or FORM or (L)	0C	014		formfeed
0F	017	⩾	SI or (O)	0F	017		shift in
11	021		DC1 or X-ON or (Q)	11	021		device control 1
12	022		DC2 or TAPE or (R)	12	022		device control 2
14	024		DC4 or TAPE or (T)	14	024		device control 4
17	027		ETB or (W)	17	027		end transmission block
18	030		CAN or CLEAR or (X)	18	030		cancel
1B	033		ESC or ESCAPE or ([)	1B	033		escape
1D	035		GS or (])	1D	035		group separator
1E	036		RS or (^)	1E	036		record separator
21	041	..		23	043	..	
22	042	—		5E	136	—	
24	044	≤		40	100	≤	
27	047	≥		3E	076	≥	
28	050	≠		22	042	≠	
2B	053	(		28	050	(	
2D	055	+		2B	053	+	
2E	056	.		2E	056	.	
30	060	0		30	060	0	
33	063	3		33	063	3	
35	065	5		35	065	5	
36	066	6		36	066	6	
39	071	9		39	071	9	
3A	072	]		5D	135	]	
3C	074	;		3B	073	;	
3F	077	\		5C	134	\	
41	101	∝		61	141	∝	
42	102	⊥		62	142	⊥	
44	104	⌊		64	144	⌊	
47	107	∇		67	147	∇	
48	110	Δ		68	150	Δ	
4B	113	`		27	047	`	
4D	115	|		6D	155	|	
4E	116	⊤		6E	156	⊤	
50	120	*		2A	052	*	
53	123	⌈		73	163	⌈	
55	125	↓		75	165	↓	
56	126	∪		76	166	∪	
59	131	↑		79	171	↑	
5A	132	⊂		7A	172	⊂	
5C	134	◇		60	140	◇	
5F	137	∧		26	046	∧	
60	140	→		71	161	→	
63	143	C		43	103	C	
65	145	E		45	105	E	
66	146	F		46	106	F	
69	151	I		49	111	I	
6A	152	J		4A	112	J	
6C	154	L		4C	114	L	
6F	157	O		4F	117	O	
71	161	Q		51	121	Q	
72	162	R		52	122	R	

TABLE A-14. CHARACTER CODE TRANSLATIONS, APL BIT-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, AND M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
74	164	T		54	124	T	
77	167	W		57	127	W	
78	170	X		58	130	X	
7B	173	⊤		6B	153	⊤	
7C	174	$		24	044	$	
7D	175	}		7D	160	}	
7E	176	÷		25	045	÷	
81	201		SOH or (A)	01	001		start of header
82	202		STX or (B)	02	002		start of text
84	204		EOT or (D)	04	004		end of transmission
87	207		BELL or (G)	07	007		bell
88	210		BS or ← or (H)	08	010		backspace
8B	213		VT or (K)	0B	013		vertical tabulate
8D	215		CR or RETURN or (M)	0D	015		carriage return
8E	216	⋘	SO or (N)	0E	016		shift out
90	220		DLE or (P)	10	020		data link escape
93	223		DC3 or X-OFF or (S)	13	023		device control 3
95	225		NAK or → or (U)	15	025		negative acknowledgement
96	226		SYN or LINE CLEAR or (V)	16	026		synchronous idle
99	231		EM or RESET or (Y)	19	031		end of medium
9A	232		SUB or ↑ or (Z)	1A	032		substitute
9C	234		FS or (\)	1C	034		file separator
9F	237		US or (−)	1F	037		unit separator
A0	240	SPACE or blank		20	040	space	
A3	243	<		3C	074	<	
A5	245	=		3D	075	=	
A6	246	≥		7C	174	≥	
A9	251	≤		21	041	≤	
AA	252	)		29	051	)	
AC	254	,		2C	054	,	
AF	257	/		2F	057	/	
B1	261	1		31	061	1	
B2	262	2		32	062	2	
B4	264	4		34	064	4	
B7	267	7		37	067	7	
B8	270	8		38	070	8	
BB	273	[		5B	133	[	
BD	275	-		2D	055	-	
BE	276	:		3A	072	:	
C0	300	←		70	160	←	
C3	303	∩		63	143	∩	
C5	305	∈		65	145	∈	
C6	306			5F	137		
C9	311	⍳		69	151	⍳	
CA	312	∘		6A	152	∘	
CC	314	□		6C	154	□	
CF	317	○		6F	157	○	
D1	321	?		3F	077	?	
D2	322	⍴		72	162	⍴	
D4	324	~		74	164	~	
D7	327	∪		77	167	∪	
D8	330	∩		78	170	∩	
DB	333	⊢		7E	176	⊢	
DD	335	{		7B	173	{	
DE	336	⍉		66	146	⍉	
E1	341	A		41	101	A	

TABLE A-14. CHARACTER CODE TRANSLATIONS, APL BIT-PAIRING CONSOLES IN TERMINAL CLASSES 1 THROUGH 3 AND 5 THROUGH 8 (M33, 713, 721, X3.64, H2000, T4014, AND M40) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	ASCII-APL Graphic	Control Character††	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
E2	342	B		42	102	B	
E4	344	D		44	104	D	
E7	347	G		47	107	G	
E8	350	H		48	110	H	
EB	353	K		4B	113	K	
ED	355	M		4D	115	M	
EE	356	N		4E	116	N	
F0	360	P		50	120	P	
F3	363	S		53	123	S	
F5	365	U		55	125	U	
F6	366	V		56	126	V	
F9	371	Y		59	131	Y	
FA	372	Z		5A	132	Z	
FF	377	■	DEL or RUBOUT	7F	177		delete

†Shown with even parity, which is the default for these terminal classes (unless PA=N or PA=I, an application program receives the same code as in normalized mode).

††A circle around a character indicates that the character key is pressed in conjunction with a CTL, CTRL, CNTRL, or CONTROL key to generate the code.

†††Shown with zero parity (eighth or uppermost bit is always zero).

TABLE A-15. CHARACTER CODE TRANSLATIONS, ASCII CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734)

Terminal ASCII†				Network ASCII (Normalized Mode Use)				
Hex. Code††	Octal Code††	Keyboard or Printer Graphic		Input or Output		Console Output Only		Graphic
		ASCII	CDC	Hex. Code†††	Octal Code†††	Hex. Code†††	Octal Code†††	
20	040	blank	blank	20	040			space
23	043	#		23	043			#
25	045	%	%	25	045			%
26	046	&		26	046			&
29	051	)	)	29	051			)
2A	052	*	*	2A	052			*
2C	054	,	,	2C	054			,
2F	057	/	/	2F	057			/
31	061	1	1	31	061			1
32	062	2	2	32	062			2
34	064	4	4	34	064			4
37	067	7	7	37	067			7
38	070	8	8	38	070			8
3B	073	;	;	3B	073			;
3D	075	=	=	3D	075			=
3E	076	>	>	3E	076			>
40	100	@	≤	40	100	60	140	@
43	103	C	C	43	103	63	143	C
45	105	E	E	45	105	65	145	E
46	106	F	F	46	106	66	146	F
49	111	I	I	49	111	69	151	I
4A	112	J	J	4A	112	6A	152	J
4C	114	L	L	4C	114	6C	154	L
4F	117	O	O	4F	117	6F	157	O
51	121	Q	Q	51	121	71	161	Q
52	122	R	R	52	122	72	162	R
54	124	T	T	54	124	74	164	T
57	127	W	W	57	127	77	167	W
58	130	X	X	58	130	78	170	X
5B	133	[	[	5B	133	7B	173	[
5D	135	]	]	5D	135	7D	175	]
5E	136	^	¬	5E	136	7E	176	^
A1	241	!		21	041			!
A2	242	"	≠	22	042			"
A4	244	$	$	24	044			$
A7	247	'		27	047			'
A8	250	(	(	28	050			(

TABLE A-15. CHARACTER CODE TRANSLATIONS, ASCII CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734) (Contd)

Terminal ASCII†				Network ASCII (Normalized Mode Use)				
Hex Code††	Octal Code††	Keyboard or Printer Graphic		Input or Output		Console Output Only		Graphic
		ASCII	CDC	Hex Code†††	Octal Code†††	Hex Code†††	Octal Code†††	
AB	253	+	+	2B	053			+
AD	255	-	-	2D	055			-
AE	256	.	.	2E	056			.
B0	260	0	0	30	060			0
B3	263	3	3	33	063			3
B5	265	5	5	35	065			5
B6	266	6	6	36	066			6
B9	271	9	9	39	071			9
BA	272	:	:	3A	072			:
BC	274	<	<	3C	074			<
BF	277	?	↓	3F	077			?
C1	301	A	A	41	101	61	141	A
C2	302	B	B	42	102	62	142	B
C4	304	D	D	44	104	64	144	D
C7	307	G	G	47	107	67	147	G
C8	310	H	H	48	110	68	150	H
CB	313	K	K	4B	113	6B	153	K
CD	315	M	M	4D	115	6D	155	M
CE	316	N	N	4E	116	6E	156	N
D0	320	P	P	50	120	70	160	P
D3	323	S	S	53	123	73	163	S
D5	325	U	U	55	125	75	165	U
D6	326	V	V	56	126	76	166	V
D9	331	Y	Y	59	131	79	171	Y
DA	332	Z	Z	5A	132	7A	172	Z
DC	334	\	≥	5C	134	7C	174	\
DF	337	_	↱	5E	135	7F	177	_

†Escape codes are not listed.

††Shown with odd parity, the only possible parity selection for these terminal classes. ASCII control codes 000 through 040_8 (without parity) are removed from input during complete editing; codes 01_8 and 03_8 (SOH and ETX, without parity) are preserved as data in full-ASCII mode, as are escape code sequences.

†††Shown with zero parity (eighth or uppermost bit is always zero). During output, codes 000 through 010_8 are converted to code 040_8 (blank); codes 012_8, 015_8, and 037_8 (LF, CR, and US) are removed. Codes for lowercase ASCII characters sent to the console are converted to the codes for the equivalent uppercase characters supported by the terminal, as shown.

TABLE A-16. CHARACTER CODE TRANSLATATIONS, ASCII CARD READERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734)

ASCII CODE BITS	1ST HEX 2ND	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL 12-0-1-8-9 space 2D	DLE 12-11-1-8-9 space 2D	SP no punch 2D	0 0 0 1B	@ 4-8 (5-8 or 11-5-8) < 3C	P 11-7 P 10	1-8 space 2D	p 12-11-7 P 10	11-0-1-8-9 space 2D	12-11-0-1-8-9 space 2D	12-0-1-9 space 2D	12-11-8-9 space 2D	12-11-0-6-9 space 2D	12-11-7-8 space 2D	12-11-0-8 space 2D	12-11-4-8-9 space 2D
0001	1	SOH 12-1-9 space 2D	DC1 11-1-9 space 2D	! 12-7-8 or 2-7-8 (11-0 or 0-2-8) ∨ 36	1 1 1 1C	A 12-1 A 01	Q 11-8 Q 11	a 12-0-1 A 01	q 12-11-8 Q 11	0-1-9 space 2D	1-9 space 2D	12-0-2-9 space 2D	11-1-8 space 2D	12-11-0-7-9 space 2D	11-0-1-8 space 2D	12-11-0-9 space 2D	12-11-5-8-9 space 2D
0010	2	STX 12-2-9 space 2D	DC2 11-2-9 space 2D	" 7-8 (4-8) ≠ 34	2 2 2 1D	B 12-2 B 02	R 11-9 R 12	b 12-0-2 B 02	r 12-11-9 R 12	0-2-9 space 2D	11-2-8-9 space 2D	12-0-3-9 space 2D	11-0-2-9 space 2D	12-11-0-8-9 space 2D	11-0-2-8 space 2D	12-11-0-2-8 space 2D	12-11-6-8-9 space 2D
0011	3	ETX 12-3-9 space 2D	DC3 11-3-9 space 2D	# 3-8 (0-6-8) ≡ 30	3 3 3 1E	C 12-3 C 0B	S 0-2 S 13	c 12-0-3 C 03	s 11-0-2 S 13	0-3-9 space 2D	3-9 space 2D	12-0-4-9 space 2D	11-0-3-9 space 2D	12-0-1-8 space 2D	11-0-3-8 space 2D	12-11-0-3-8 space 2D	12-11-7-8-9 space 2D
0100	4	EOT 7-9 space 2D	DC4 4-8-9 space 2D	$ 11-3-8 $ 2B	4 4 4 1F	D 12-4 D 04	T 0-3 T 14	d 12-0-4 D 04	t 11-0-3 T 14	0-4-9 space 2D	4-9 space 2D	12-0-5-9 space 2D	11-0-4-9 space 2D	12-0-2-8 space 2D	11-0-4-8 space 2D	12-11-0-4-8 space 2D	11-0-2-8-9 space 2D
0101	5	ENQ 0-5-8-9 space 2D	NAK 5-8-9 space 2D	% 0-4-8 (6-8) % 33	5 5 5 20	E 12-5 E 05	U 0-4 U 15	e 12-0-5 E 05	u 11-0-4 U 15	11-5-9 space 2D	5-9 space 2D	12-0-6-9 space 2D	11-0-5-9 space 2D	12-0-3-8 space 2D	11-0-5-8 space 2D	12-11-0-5-8 space 2D	11-0-3-8-9 space 2D
0110	6	ACK 0-6-8-9 space 2D	SYN 2-9 space 2D	& 12 (0-7-8) ^ 37	6 6 6 21	F 12-8 F 06	V 0-5 V 16	f 12-0-6 F 06	v 11-0-5 V 16	12-6-9 space 2D	6-9 space 2D	12-0-7-9 space 2D	11-0-6-9 space 2D	12-0-4-8 space 2D	11-0-6-8 space 2D	12-11-0-6-8 space 2D	11-0-4-8-9 space 2D
0111	7	BEL 0-7-8-9 space 2D	ETB 0-8-9 space 2D	' 5-8 (11-5-8) ' 38	7 7 7 22	G 12-7 G 07	W 0-6 W 17	g 12-0-7 G 07	w 11-0-6 W 17	11-7-9 space 2D	12-8-9 space 2D	12-0-8-9 space 2D	11-0-7-9 space 2D	12-0-5-8 space 2D	11-0-7-8 space 2D	12-11-0-7-8 space 2D	11-0-5-8-9 space 2D
1000	8	BS 11-6-9 space 2D	CAN 11-8-9 space 2D	(12-5-8 (0-4-8) (29	8 8 8 23	H 12-8 H 08	X 0-7 X 18	h 12-0-8 H 08	x 11-0-7 X 18	0-8-9 space 2D	8-9 space 2D	12-1-8 space 2D	11-0-8-9 space 2D	12-0-6-8 space 2D	12-11-0-1-8 space 2D	12-0-2-8-9 space 2D	11-0-6-8-9 space 2D
1001	9	HT 12-0-5 space 2D	EM 11-1-8-9 space 2D	) 11-5-8 (12-4-8)) 2A	9 9 9 24	I 12-9 I 09	Y 0-8 Y 19	i 12-0-9 I 09	y 11-0-8 Y 19	0-1-8-9 space 2D	1-8-9 space 2D	12-11-1-9 space 2D	0-1-8 space 2D	12-0-7-8 space 2D	12-11-0-1 space 2D	12-0-3-8-9 space 2D	11-0-7-8-9 space 2D
1010	A	LF 0-5-9 space 2D	SUB 7-8-9 space 2D	* 11-4-8 * 27	: 2-8 : 00	J 11-1 J 0A	Z 0-9 Z 1A	j 12-11-1 J 0A	z 11-0-9 Z 1A	0-2-8-9 space 2D	2-8-9 space 2D	12-11-2-9 space 2D	12-11-0 space 2D	12-11-1-8 space 2D	12-11-0-2 space 2D	12-0-4-8-9 space 2D	12-11-0-2-8-9 space 2D
1011	B	VT 12-3-8-9 space 2D	ESC 0-7-9 space 2D	+ 12-6-8 (12) + 25	; 11-6-8 (12-7-8) ; 3F	K 11-2 K 0B	| 12-0 or 12-2-8 (7-8 or 12-0 or 12-3-8) | 31	k 12-11-2 K 0B	12-0 space 2D	0-3-8-9 space 2D	3-8-9 space 2D	12-11-3-9 space 2D	12-11-0-1-9 space 2D	12-11-2-8 space 2D	12-11-0-3 space 2D	12-0-5-8-9 space 2D	12-11-0-3-8-9 space 2D
1100	C	FF 12-4-8-9 space 2D	FS 11-4-8-9 space 2D	, 0-3-8 , 2E	< 12-4-8 (12-0 or 5-8) < 3A	L 11-3 L 0C	\ 0-2-8 or 0-5-8 (12-5-8) > 3D	l 12-11-3 L 0C	12-11 space 2D	0-4-8-9 space 2D	12-4-9 space 2D	12-11-4-9 space 2D	12-11-0-2-9 space 2D	12-11-3-8 space 2D	12-11-0-4 space 2D	12-0-6-8-9 space 2D	12-11-0-4-8-9 space 2D
1101	D	CR 12-5-8-9 space 2D	GS 11-5-8-9 space 2D	- 11 - 26	= 6-8 (3-8) = 2C	M 11-4 M 0D	] 11-0 or 11-2-8 (11-0 or 0-2-8)] 32	m 12-11-4 M 0D	11-0 space 2D	12-1-8-9 space 2D	11-4-9 space 2D	12-11-5-9 space 2D	12-11-0-3-9 space 2D	12-11-4-8 space 2D	12-11-0-5 space 2D	12-0-7-8-9 space 2D	12-11-0-5-8-9 space 2D
1110	E	SO 12-6-8-9 space 2D	RS 11-6-8-9 space 2D	. 12-3-8 . 2F	> 0-6-8 (11-7-8) > 3B	N 11-5 N 0E	^ 11-7-8 (12-6-8) ¬ 3E	n 12-11-5 N 0E	11-0-1 space 2D	12-2-8-9 space 2D	.6-8-9 space 2D	12-11-6-9 space 2D	12-11-0-4-9 space 2D	12-11-5-8 space 2D	12-11-0-6 space 2D	12-11-2-8-9 space 2D	12-11-0-6-8-9 space 2D
1111	F	SI 12-7-8-9 space 2D	US 11-7-8-9 space 2D	/ 0-1 / 28	? 0-7-8 (11-6-8) ↓ 39	O 11-6 O 0F	0-5-8 ↱ 35	o 12-11-8 O 0F	DEL 12-7-9 space 2D	11-3-8-9 space 2D	11-0-1-9 space 2D	12-11-7-9 space 2D	12-11-0-5-9 space 2D	12-11-6-8 space 2D	12-11-0-7 space 2D	12-11-3-8-9 space 2D	12-11-0-7-8-9 space 2D

LEGEND: ASCII Character → + ; 12-6-8 (12) → 029 Card Code, 026 Card Code
CDC Graphic Character → + 25 → Display Code (Hexadecimal)

NOTE: Not all Mode 4A emulators support all of the punch patterns shown.

TABLE A-17. CHARACTER CODE TRANSLATIONS, EXTERNAL BINARY CODED (BCD) CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734)

Terminal External BCD†				Network ASCII (Normalized Mode Use)				
Hex. Code††	Octal Code††	Keyboard or Printer Graphic		Input or Output		Console Output Only		Graphic
		ASCII	CDC	Hex. Code†††	Octal Code†††	Hex. Code†††	Octal Code†††	
10	020	:	:	3A	072			:
20	040	-	-	2D	055			-
23	043	L	L	4C	114	6C	154	L
25	045	N	N	4E	116	6E	156	N
26	046	O	O	4F	117	6F	157	O
29	051	R	R	52	122	72	162	R
2A	052	!	∨	21	041			!
2C	054	*	*	2A	052			*
2F	057	>	>	3E	076			>
31	061	A	A	41	101	61	141	A
32	062	B	B	42	102	62	142	B
34	064	D	D	44	104	64	144	D
37	067	G	G	47	107	67	147	G
38	070	H	H	48	110	68	150	H
3B	073	.	.	2E	056			.
3D	075			5C	134	7C	174	\
43	103	3	3	33	063			3
45	105	5	5	35	065			5
46	106	6	6	36	066			6
49	111	9	9	39	071			9
4A	112	0	0	30	060			0
4C	114	=	≠	22	042			"
4F	117	[	[	5B	133	7B	173	[
51	121	/	/	2F	057			/
52	122	S	S	53	123	73	163	S
54	124	U	U	55	125	75	165	U
57	127	X	X	58	130	78	170	X
58	130	Y	Y	59	131	79	171	Y
5B	133	,	,	2C	054			,
5D	135	_	↱	5F	137	7F	177	_
5E	136	#	≡	23	043			#
A1	241	J	J	4A	112	6A	152	J
A2	242	K	K	4B	113	6B	153	K
A4	244	M	M	4D	115	6D	155	M
A7	247	P	P	50	120	70	160	P
A8	250	Q	Q	51	121	71	161	Q
AB	253	$	$	24	044			$

TABLE A-17. CHARACTER CODE TRANSLATIONS, EXTERNAL BINARY CODED (BCD) CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734) (Contd)

Terminal External BCD†				Network ASCII (Normalized Mode Use)				
Hex. Code††	Octal Code††	Keyboard or Printer Graphic		Input or Output		Console Output Only		Graphic
		ASCII	CDC	Hex. Code†††	Octal Code†††	Hex. Code†††	Octal Code†††	
AD	255	'	↑	27	047			,
AE	256	?	↓	3F	077			?
B3	263	C	C	43	103	63	143	C
B5	265	E	E	45	105	65	145	E
B6	266	F	F	46	106	66	146	F
B9	271	I	I	49	111	69	151	I
BA	272	<	<	3C	074			<
BC	274	)	)	29	051			)
BF	277	;	;	3B	073			;
C1	301	1	1	31	061			1
C2	302	2	2	32	062			2
C4	304	4	4	34	064			4
C7	307	7	7	37	067			7
C8	310	8	8	38	070			8
CB	313	=	=	3D	075			=
CD	315	@	≤	40	100	60	140	@
CE	316	%	%	25	045			%
D0	320	blank	blank	20	040			space
D3	323	T	T	54	124	74	164	T
D5	325	V	V	56	126	76	166	V
D6	326	W	W	57	127	77	167	W
D9	331	Z	Z	5A	132	7A	172	Z
DA	332	]	]	5D	135	7D	175	]
DC	334	(	(	28	050			(
DF	337	&	∧	26	046			&
D0	320	∧ or blank	¬ or ■ or none			5E, 7E	136, 176	∧§

†Escape codes and control codes are not listed.

††Shown with odd parity, the only possible parity selection for these terminal classes.

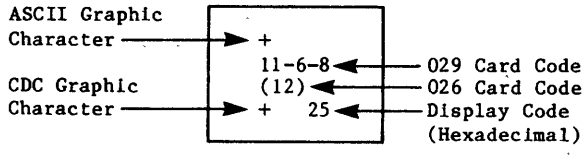
†††Shown with zero parity (eighth or uppermost bit is always zero). During output, codes 000 through 037_8 are converted to code 320_8 (blank). Codes for lowercase ASCII characters sent to the console are converted to the codes for the equivalent uppercase characters supported by the terminal, as shown.

§Input and output of this symbol is not possible on some terminals. BCD transmission conventions support the rubout symbol ■ as an internal terminal memory parity error indicator instead. The ASCII codes 136_8 and 176_8 are output as a blank.

TABLE A-18. CHARACTER CODE TRANSLATIONS, EXTERNAL BINARY CODED DECIMAL (BCD) CARD READERS IN TERMINAL CLASSES 10 AND 15 (200UT AND 734)

BCD BITS	1ST HEX 2ND	0010 2	0011 3	0100 4	0101 5	BCD BITS	1ST HEX 2ND	0010 2	0011 3	0100 4	0101 5
0000	0	- 11 - 26	+ 12-6-8 (12) + 25	: 2-8 : 00	SP no punch space 2D	1000	8	Q 11-8 Q 11	H 12-8 H 08	8 8 8 23	Y 0-8 Y 19
0001	1	J 11-1 J 0A	A 12-1 A 01	1 1 1 1C	/ 0-1 / 28	1001	9	R 11-9 R 12	I 12-9 I 09	9 9 9 24	Z 0-9 Z 1A
0010	2	K 11-2 K 0B	B 12-2 B 02	2 2 2 1D	S 0-2 S 13	1010	A	! 12-7-8 (11-0) ∨ 36	< 12-4-8 (12-0) < 3A	0 0 0 1B	] 11-2-8 (0-2-8)] 32
0011	3	L 11-3 L 0C	C 12-3 C 03	3 3 3 1E	T 0-3 T 14	1011	B	$ 11-3-8 $ 2B	. 12-3-8 . 2F	= 6-8 (3-8) = 2C	, 0-3-5 , 2E
0100	4	M 11-4 M 0D	D 12-4 D 04	4 4 4 1F	U 0-4 U 15	1100	C	* 11-4-8 * 27	) 11-5-8 (12-4-8)) 2A	" 5-8 or 7-8 (4-8) ≠ 34	(12-5-8 (0-4-8) (29
0101	5	N 11-5 N 0E	E 12-5 E 05	5 5 5 20	V 0-5 V 16	1101	D	′ 12 or 5-8 (11-5-8) ↑ 38	\ 0-2-8 (12-5-8) ≥ 5C	@ 11-7-8 or 4-8 (5-8) < 3C	$\overline{0}$ 0-5-8 ↱ 35
0110	6	O 11-6 O 0F	F 12-6 F 06	6 6 6 21	W 0-6 W 17	1110	E	? 12-7-8 or 0-7-8 (11-6-8) ↓ 39	∧ 11-7-8 (12-6-8) ¬ 3E	% 0-4-8 (6-8) % 33	# 3-8 (0-6-8) ≡ 30
0111	7	P 11-7 P 10	G 12-7 G 07	7 7 7 22	X 0-7 X 18	1111	F	> 0-6-8 (11-7-8) > 3B	; 11-6-8 (12-7-8) ; 3F	[4-8 or 12-2-8 (7-8) [31	& 0-7-8 or 12 (0-7-8) ∧ 37

LEGEND:



NOTE:

Not all Mode 4A emulators support the alternate punch patterns shown.

TABLE A-19. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 11, 12, AND 13 (711, 714, AND 714X)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character§
01	001		SOH or (A)	01	001		start of header§§
02	002		STX or (B)	20	040	space	
04	004		EOT or (D)	20	040	space	
07	007		BELL or (G)	20	040	space	
08	010		BS or ← or (H)	20	040	space	
0B	013		VT or (K)	0B	013		vertical tabulate
0D	015		CR or RETURN or (M)				
0E	016		SO or (N)	0E	016		shift out
10	020		DLE or (P)	10	020		data link escape
13	023		DC3 or X-OFF or (S)	13	023		device control 3
15	025		NAK or → or (U)	15	025		negative acknowledgment
16	026		SYN or LINE CLEAR or (V)	16	026		synchronous idle
19	031		EM or RESET or (Y)	19	031		end of medium
1A	032		SUB or ↑ or (Z)	1A	032		substitute
1C	034		FS or (\)	1C	034		file separator
1F	037		US or (−)	20	040	space	
20	040	SPACE or blank		20	040	space	
23	043	#		23	043	#	
25	045	%		25	045	%	
26	046	&		26	046	&	
29	051	)		29	051	)	
2A	052	*		2A	052	*	
2C	054	,		2C	054	,	
2F	057	/		2F	057	/	
31	061	1		31	061	1	
32	062	2		32	062	2	
34	064	4		34	064	4	
37	067	7		37	067	7	
38	070	8		38	070	8	
3B	073	;		3B	073	;	
3D	075	=		3D	075	=	
3E	076	>		3E	076	>	
40	100	@		40	100	@	
43	103	C		43	103	C	
45	105	E		45	105	E	
46	106	F		46	106	F	
49	111	I		49	111	I	
4A	112	J		4A	112	J	
4C	114	L		4C	114	L	
4F	117	O		4F	117	O	
51	121	Q		51	121	Q	
52	122	R		52	122	R	
54	124	T		54	124	T	
57	127	W		57	127	W	
58	130	X		58	130	X	
5B	133	[		5B	133	[	
5D	135	]		5D	135	]	
5E	136	∧ or ¬		5E	136	∧	
61	141	a		61	141	a	
62	142	b		62	142	b	
64	144	d		64	144	d	
67	147	g		67	147	g	
68	150	h		68	150	h	
6B	153	k		6B	153	k	
6D	155	m		6D	155	m	
6E	156	n		6E	156	n	
70	160	p		70	160	p	

TABLE A-19. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 11, 12, AND 13 (711, 714, AND 714X) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character§
73	163	s		73	163	s	
75	165	u		75	165	u	
76	166	v		76	166	v	
79	171	y		79	171	y	
7A	172	z		7A	172	z	
7C	174	¦ or ↑ or |		7C	174	¦	
7F	177	■	DEL or RUBOUT	7F	177		delete
80	200		NUL or (@)	20	040	space	
83	203		ETX or (C)	03	003		end of text§§
85	205		ENQ or WRU or (E)	20	040	space	
86	206		ACK or RU or (F)	20	040	space	
89	211		HT or (I)	09	011		horizontal tabulate
8A	212		LF or NL or ↓ or (J) or NEW LINE	0A	012		linefeed
8C	214		FF or FORM or (L)	0C	014		formfeed
8F	217		SI or (O)	0F	017		shift in
91	221		DC1 or X-ON or (Q)	11	021		device control 1
92	222		DC2 or TAPE or (R)	12	022		device control 2
94	224		DC4 or TAPE or (T)	14	024		device control 4
97	227		ETB or (W)	17	027		end transmission block
98	230		CAN or CLEAR or (X)	18	030		cancel
9B	233		ESC or ESCAPE or ([)	1B	033		escape
9D	235		GS or (])	1D	035		group separator
9E	236		RS or (^)	1E	036		record separator
A1	241	!		21	041	!	
A2	242	"		22	042	"	
A4	244	$		24	044	$	
A7	247	'		27	047	'	
A8	250	(		28	050	(	
AB	253	+		2B	053	+	
AD	255	-		2D	055	-	
AE	256	.		2E	056	.	
B0	260	0		30	060	0	
B3	263	3		33	063	3	
B5	265	5		35	065	5	
B6	266	6		36	066	6	
B9	271	9		39	071	9	
BA	272	:		3A	072	:	
BC	274	<		3C	074	<	
BF	277	?		3F	077	?	
C1	301	A		41	101	A	
C2	302	B		42	102	B	
C4	304	D		44	104	D	
C7	307	G		47	107	G	
C8	310	H		48	110	H	
CB	313	K		4B	113	K	
CD	315	M		4D	115	M	
CE	316	N		4E	116	N	
D0	320	P		50	120	P	
D3	323	S		53	123	S	
D5	325	U		55	125	U	
D6	326	V		56	126	V	
D9	331	Y		59	131	Y	
DA	332	Z		5A	132	Z	
DC	334	\		5C	134	∨	
DF	337	_ or ←		5F	137		
E0	340	‾		60	140	‾	
E3	343	c		63	143	c	

TABLE A-19. CHARACTER CODE TRANSLATIONS, CONSOLES AND LINE PRINTERS IN TERMINAL CLASSES 11, 12, AND 13 (711, 714, AND 714X) (Contd)

Terminal ASCII (Transparent Mode Use)				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	ASCII Graphic	Control Character††	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character§
E5	345	e		65	145	e	
E6	346	f		66	146	f	
E9	351	i		69	151	i	
EA	352	j		6A	152	j	
EC	354	l		6C	154	l	
EF	357	o		6F	157	o	
F1	361	q		71	161	q	
F2	362	r		72	162	r	
F4	364	t		74	164	t	
F7	367	w		77	167	w	
F8	370	x		78	170	x	
FB	373	{		7B	173	{	
FD	375	}		7D	175	}	
FE	376	~ or ¬		7E	176	~	

†Shown with odd parity, which is the default for these terminal classes (unless PA=N or PA=I, an application program receives the same code as in normalized mode).

††A circle around a character indicates that the character key is pressed in conjunction with a CTL, CTRL, CNTRL, or CONTROL key to generate the code.

†††Shown with zero parity (eighth or uppermost bit is always zero).

§Converted to a space (040_8) within a batch printer file.

§§Converted to a space (040_8) during complete editing.

TABLE A-20. ASCII CHARACTER CODE TRANSLATIONS, EBCD CONSOLES IN TERMINAL CLASS 4 (2741)

Terminal EBCD				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	EBCD Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
01	001	_ or -		5F or 2D	137 or 055	_ or -	
02	002	¢ or @		21 or 40	140 or 100	` or @	
04	004	* or 8		2A or 38	052 or 070	* or 8	
07	007	H or h		48 or 68	110 or 150	H or h	
08	010	: or 4		3A or 34	072 or 064	: or 4	
0B	013	D or d		44 or 64	104 or 144	D or d	
0D	015		RES or RESTORE	00	000		null
0E	016		BY or BYPASS	00	000		null
10	020	< or 2		3C or 32	074 or 062	< or 2	
13	023	B or b		42 or 62	102 or 142	B or b	
15	025		undefined	00	000		null
16	026		undefined	00	000		null
19	031	O or o		4F or 6F	117 or 157	O or o	
1A	032	W or w		57 or 77	127 or 167	W or w	
1C	034		UCS or UPPERCASE	0E	016		shift out§
1F	037		LCS or LOWERCASE	0F	017		shift in§
20	040	= or 1		3D or 31	075 or 061	= or 1	
23	043	A or a		41 or 61	101 or 141	A or a	
25	045	R or r		52 or 72	122 or 162	R or r	
26	046	Z or z		5A or 7A	132 or 172	Z or z	
29	051	N or n		4E or 6E	116 or 156	N or n	
2A	052	V or v		56 or 76	126 or 166	V or v	
2C	054		RO or READER STOP	14	024		device control 4
2F	057		HT or TAB	09	011		horizontal tabulate
31	061	L or 1		4C or 6C	114 or 154	L or 1	
32	062	T or t		54 or 74	124 or 164	T or t	
34	064	" or #		22 or 23	042 or 043	= or #	
37	067	¬ or .		5E or 2E	136 or 056	^ or .	
38	070	> or 7		3E or 37	076 or 067	> or 7	
3B	073	G or g		47 or 67	107 or 147	G or g	
3D	075		IL or IDLE or NULL	00	000		null
3E	076		PRE or PREFIX	01	001		start of header§
40	100	space		20	040	space	
43	103	+ or &		2B or 26	053 or 046	+ or &	
45	105	Q or q		51 or 71	121 or 161	Q or q	
46	106	Y or y		59 or 79	131 or 171	Y or y	
49	111	M or m		4D or 6D	115 or 155	M or m	
4A	112	U or u		55 or 75	125 or 165	U or u	
4C	114		PN or PUNCH ON	11	021		device control 1 (tape on)
4F	117		PF or PUNCH OFF	13	023		device control 3 (tape off)
51	121	K or k		4B or 6B	113 or 153	K or k	
52	122	S or s		53 or 73	123 or 163	S or s	
54	124	) or 0		29 or 30	051 or 060	) or 0	
57	127		undefined	00	000		null
58	130	' or 6		27 or 36	047 or 066	' or 6	
5B	133	F or f		46 or 66	106 or 146	F or f	
5D	135		BS or BACKSPACE	08	010		backspace
5E	136		EOB	17	027		end transmission block§
61	141	J or j		4A or 6A	112 or 152	J or j	
62	142	? or /		3F or 2F	077 or 057	? or /	
64	144	(or 9		28 or 39	050 or 071	(or 9	
67	147	I or i		49 or 69	111 or 151	I or i	
68	150	% or 5		25 or 35	045 or 065	% or 5	
6B	153	E or e		45 or 65	105 or 145	E or e	
6D	155		NL or CR or RETURN	0D	015		carriage return
6E	156		LF or LINE FEED	0A	012		linefeed

TABLE A-20. ASCII CHARACTER CODE TRANSLATIONS, EBCD CONSOLES IN TERMINAL CLASS 4 (2741) (Contd)

Terminal EBCD				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	EBCD Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
70	160	; or 3		3B or 33	073 or 063	; or 3	
73	163	C or c		43 or 63	103 or 143	C or c	
75	165	! or $		21 or 24	041 or 044	! or $	
76	166	| or ,		7C or 2C	174 or 054	¦ or ,	
79	171	P or p		50 or 70	120 or 160	P or p	
7A	172	X or x		58 or 78	130 or 170	X or x	
7C	174		EOT	04	004		end of transmission§
7F	177		DEL	7F	177		delete
00	000	space		5B thru 5D	133 thru 135	[or \ or]	
00	000	space		60	140	`	
00	000	space		7B	173	{	
00	000	space		7D or 7E	175 or 176	} or ~	
3D	075		IL or IDLE or NULL§§	02	002		start of text
3D	075		IL or IDLE or NULL§§	03	003		end of text
3D	075		IL or IDLE or NULL§§	05	005		enquire
3D	075		IL or IDLE or NULL§§	07	007		bell
3D	075		IL or IDLE or NULL§§	0B or 0C	013 or 014		vertical tabulate or formfeed
3D	075		IL or IDLE or NULL§§	10	020		data link escape
3D	075		IL or IDLE or NULL§§	12	022		device control 2
3D	075		IL or IDLE or NULL§§	14 thru 16	024 thru 026		device control 4, negative acknowledge, or synchronize
3D	075		IL or IDLE or NULL§§	18 thru 1F	030 thru 037		cancel, end of media, substitute, escape, file separator, group separator, record separator, or unit separator

†Shown with odd parity; odd parity is the default for this terminal class.

††Each input line is assumed to begin in lowercase. Input characters are translated to lowercase ASCII characters unless prefixed by the UCS code. Once a case shift occurs, it remains in effect until another case shift code is received, the page width is reached, or the line is transmitted to the host computer. During output, case is preserved by insertion of case shift codes where needed.

†††Shown with zero parity (eighth or uppermost bit is always zero).

§Not transmitted to the host computer after translation during input.

§§Output translation only.

TABLE A-21. APL CHARACTER CODE TRANSLATIONS, EBCD CONSOLES IN TERMINAL CLASS 4 (2741)

Terminal EBCD-APL				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	EBCD-APL Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
01	001	_ or +		5F or 2D	137 or 053	_ or +	
02	002	→ or ←		71 or 70	161 or 160	→ or ←	
04	004	≠ or 8		22 or 38	042 or 070	≠ or 8	
07	007	Δ or H		68 or 48	150 or 110	Δ or H	
08	010	≤ or 4		40 or 34	100 or 064	≤ or 4	
0B	013	⌊ or D		64 or 44	144 or 104	⌊ or D	
0D	015		undefined	00	000		null
0E	016		undefined	00	000		null
10	020	- or 2		2D or 32	055 or 062	- or 2	
13	023	⊥ or B		42 or 62	142 or 102	⊥ or B	
15	025		undefined	00	000		null
16	026		undefined	00	000		null
19	031	○ or O		6F or 4F	157 or 117	○ or O	
1A	032	ω or W		77 or 57	167 or 127	ω or W	
1C	034		UCS or UPPERCASE	0E	016		shift out§
1F	037		LCS or LOWERCASE	0F	017		shift in§
20	040	" or 1		22 or 31	042 or 061	" or 1	
23	043	⍺ or A		61 or 41	141 or 101	⍺ or A	
25	045	⍴ or R		72 or 52	162 or 122	⍴ or R	
26	046	⊂ or Z		7A or 5A	172 or 132	⊂ or Z	
29	051	⍵ or N		6E or 4E	156 or 116	⍵ or N	
2A	052	∪ or V		76 or 56	166 or 126	∪ or V	
2C	054		undefined	00	000		null
2F	057		HT or TAB	06	006		horizontal tabulate
31	061	□ or L		6C or 4C	154 or 114	□ or L	
32	062	~ or T		74 or 54	164 or 124	~ or T	
34	064	) or]		29 or 5D	051 or 135	) or]	
37	067	: or .		3A or 2E	072 or 056	: or .	
38	070	> or 7		3E or 37	076 or 067	> or 7	
3B	073	∇ or G		67 or 47	147 or 107	∇ or G	
3D	075		IL or IDLE or NULL	00	000		null
3E	076		PRE or PREFIX	1B	033		escape
40	100	space		20	040	space	
43	103	+ or X		25 or 66	045 or 146	+ or X	
45	105	? or Q		3F or 51	077 or 121	? or Q	
46	106	↑ or Y		79 or 59	171 or 131	↑ or Y	
49	111	| or M		6D or 4D	155 or 115	| or M	
4A	112	↓ or U		75 or 55	165 or 125	↓ or U	
4C	114		undefined	00	000		null
4F	117		undefined	00	000		null
51	121	⊣ or K		6B or 4B	153 or 113	⊣ or K	
52	122	⊢ or S		73 or 53	163 or 123	⊢ or S	
54	124	∧ or 0		26 or 30	046 or 060	∧ or 0	
57	127		undefined	00	000		null
58	130	≥ or 6		7C or 36	174 or 066	≥ or 6	
5B	133	≡ or F		5E or 46	136 or 106	≡ or F	
5D	135		BS or BACKSPACE	08	010		backspace
5E	136		EOB	17	027		end transmission block§
61	141	∘ or J		6A or 4A	152 or 112	∘ or J	
62	142	\ or /		5C or 2F	134 or 057	\ or /	
64	144	⍱ or 9		21 or 39	041 or 071	⍱ or 9	
67	147	⍳ or I		69 or 49	151 or 111	⍳ or I	
68	150	= or 5		3D or 35	075 or 065	= or 5	
6B	153	∈ or E		65 or 45	145 or 105	∈ or E	
6D	155		NL or CR or RETURN	0D	015		carriage return
6E	156		LF or LINE FEED	0A	012		line feed
70	160	< or 3		3C or 33	074 or 063	< or 3	
73	163	∩ or C		63 or 43	143 or 103	∩ or C	
75	165	(or [		28 or 5B	050 or 133	(or [	
76	166	; or ,		3B or 2C	073 or 054	; or ,	
79	171	* or P		2A or 50	052 or 120	* or P	
7A	172	⊃ or X		78 or 58	170 or 130	⊃ or X	

TABLE A-21. APL CHARACTER CODE TRANSLATIONS, EBCD CONSOLES IN TERMINAL CLASS 4 (2741) (Contd)

Terminal EBCD-APL				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	EBCD-APL Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
7C	174		EOT	04	004		end of transmission§
7F	177		DEL	7F	177		delete
00	000	space§§		27	047	'	
00	000	space§§		60	140	◇	
00	000	space§§		7B	173	{	
00	000	space§§		7D	175	}	
3D	075		IL or IDLE or NULL§§	02	002		start of text
3D	075		IL or IDLE or NULL§§	03	003		end of text
3D	075		IL or IDLE or NULL§§	05	005		enquire
3D	075		IL or IDLE or NULL§§	07	007		bell
3D	075		IL or IDLE or NULL§§	0B or 0C	013 or 014		vertical tabulate or form feed
3D	075		IL or IDLE or NULL§§	10 thru 16	020 thru 026		data link escape, device control 1 thru device control 4, negative acknowledge, or synchronize
3D	075		IL or IDLE or NULL§§	18 thru 1F	030 thru 037		cancel, end of media, substitute, escape file separator, group separator, record separator, or unit separator

†Shown with odd parity; odd parity is the default for this terminal class.

††Each input line is assumed to begin in lowercase. Input characters are translated to lowercase ASCII characters unless prefixed by the UCS code. Once a case shift occurs, it remains in effect until another case shift code is received, the page width is reached, or the line is transmitted to the host computer. During output, case is preserved by insertion of case shift codes where needed.

†††Shown with zero parity (eighth or uppermost bit is always zero).

§Not transmitted to the host computer after translation during input.

§§Output translation only.

TABLE A-22. ASCII CHARACTER CODE TRANSLATIONS, CORRESPONDENCE CODE CONSOLES IN TERMINAL CLASS 4 (2741)

Terminal Correspondence Code				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	Correspondence Code Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
01	001	1/4 or 1/2		5B or 5D	137 or 135	[or]	
02	002	T or t		54 or 74	124 or 164	T or t	
04	004	$ or 4		24 or 34	044 or 064	$ or 4	
07	007	? or /		3F or 2F	077 or 057	? or /	
08	010	% or 5		25 or 35	045 or 065	% or 5	
0B	013	P or p		50 or 70	120 or 160	P or p	
0D	015		RES or RESTORE	00	000		null
0E	016		BY or BYPASS	00	000		null
10	020	@ or 2		40 or 32	100 or 062	@ or 2	
13	023	+ or =		2B or 3D	053 or 075	+ or =	
15	025		undefined	00	000		null
16	026		undefined	00	000		null
19	031	I or i		49 or 69	111 or 151	I or i	
1A	032	K or k		4B or 6B	113 or 153	K or k	
1C	034		UCS or UPPERCASE	0E	016		shift out§
1F	037		LCS or LOWERCASE	0F	017		shift in§
20	040	± or 1		7C or 31	174 or 061	¦ or 1	
23	043	G or g		47 or 67	107 or 147	G or g	
25	045	S or s		53 or 73	123 or 163	S or s	
26	046	H or h		48 or 68	110 or 150	H or h	
29	051	R or r		52 or 72	122 or 162	R or r	
2A	052	D or d		44 or 64	104 or 144	D or d	
2C	054		RO or READER STOP	14	024		device control 4
2F	057		HT or TAB	09	011		horizontal tabulate
31	061	V or v		56 or 76	126 or 166	V or v	
32	062	U or u		55 or 75	125 or 165	U or u	
34	064	(or 9		28 or 39	050 or 071	(or 9	
37	067	_ or -		5F or 2D	137 or 055	_ or -	
38	070	¯ or 8		2A or 38	052 or 070	¯ or 8	
3B	073	,		2C	054	,	
3D	075		IL or IDLE or NULL	00	000		null
3E	076		PRE or PREFIX	1B	033		escape
40	100	space		20	040	space	
43	103	J or j		4A or 6A	112 or 152	J or j	
45	105	O or o		4F or 6F	117 or 157	O or o	
46	106	L or l		4C or 6C	114 or 154	L or l	
49	111	" or '		22 or 27	042 or 041	" or '	
4A	112	E or e		45 or 65	105 or 145	E or e	
4C	114		PN or PUNCH ON	11	021		device control 1 (tape on)
4F	117		PF or PUNCH OFF	13	023		device control 3 (tape off)
51	121	.		2E	056	.	
52	122	N or n		4E or 6E	116 or 156	N or n	
54	124	Z or z		5A or 7A	132 or 172	Z or z	
57	127		undefined	00	000		null
58	130	¢ or 6		21 or 36	041 or 066	! or 6	
5B	133	Q or q		51 or 71	121 or 161	Q or q	
5D	135		BS or BACKSPACE	08	010		backspace
5E	136		EOB	17	027		end transmission block§
61	141	M or m		4D or 6D	115 or 155	M or m	
62	142	X or x		58 or 78	130 or 170	X or x	
64	144	) or 0		29 or 30	051 or 060	) or 0	
67	147	Y or y		79 or 59	131 or 171	Y or y	
68	150	& or 7		26 or 37	046 or 067	& or 7	
6B	153	: or ;		3A or 3B	072 or 073	: or ;	
6D	155		NL or CR or RETURN	0D	015		carriage return
6E	156		LF or LINE FEED	0A	012		line feed
70	160	# or 3		23 or 33	043 or 063	# or 3	
73	163	F of f		46 or 66	106 or 146	F or f	
75	165	W or w		57 or 77	127 or 167	W or w	

TABLE A-22. ASCII CHARACTER CODE TRANSLATIONS, CORRESPONDENCE CODE CONSOLES IN TERMINAL CLASS 4 (2741) (Contd)

Terminal Correspondence Code				Network ASCII (Normalized Mode Use)			
Hex. Code†	Octal Code†	Correspondence Code Graphic††	Control Character	Hex. Code†††	Octal Code†††	ASCII Graphic	Control Character
76	166	B or b		42 or 62	102 or 142	B or b	
79	171	A or a		41 or 61	101 or 141	A or a	
7A	172	C or c		43 or 63	103 or 143	C or c	
7C	174		EOT	04	004		end of transmission§
7F	177		DEL	18	030		cancel
00	000	space§§		27	047	'	
00	000	space§§		5C	134	\	
00	000	space§§		5E	136	^	
00	000	space§§		60	140	`	
00	000	space§§		7B	173	{	
00	000	space§§		7D or 7E	175 or 176	} or ~	
3D	075		IL or IDLE or NULL§§	01	001		start of header
3D	075		IL or IDLE or NULL§§	02	002		start of text
3D	075		IL or IDLE or NULL§§	03	003		end of text
3D	075		IL or IDLE or NULL§§	05	005		enquire
3D	075		IL or IDLE or NULL§§	07	007		bell
3D	075		IL or IDLE or NULL§§	0B or 0C	013 or 014		vertical tabulate or form feed
3D	075		IL or IDLE or NULL§§	10	020		data link escape
3D	075		IL or IDLE or NULL§§	12	022		device control 2
3D	075		IL or IDLE or NULL§§	14 thru 16	024 thru 026		device control 4, negative acknowledge, or synchronize
3D	075		IL or IDLE or NULL§§	18 thru 1F	030 thru 037		cancel, end of media, substitute, file separator, group separator, record separator, or unit separator

†Shown with odd parity; odd parity is the default for this terminal class.

††Each input line is assumed to begin in lowercase. Input characters are translated to lowercase ASCII characters unless prefixed by the UCS code. Once a case shift occurs, it remains in effect until another case shift code is received, the page width is reached, or the line is transmitted to the host computer. During output, case is preserved by insertion of case shift codes where needed.

†††Shown with zero parity (eighth or uppermost bit is always zero).

§Not transmitted to the host computer after translation during input.

§§Output translation only.

TABLE A-23. APL CHARACTER CODE TRANSLATIONS, CORRESPONDENCE CODE CONSOLES IN TERMINAL CLASS 4 (2741)

Terminal Correspondence Code				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	Correspondence Code APL Graphic††	Control Character	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
01	001	→ or ←		71 or 70	161 or 160	→ or ←	
02	002	~ or T		74 or 54	164 or 124	~ or T	
04	004	≤ or 4		40 or 34	100 or 064	≤ or 4	
07	007	\ or /		5C or 2F	134 or 057	\ or /	
08	010	= or 5		3D or 35	075 or 065	= or 5	
0B	013	* or P		2A or 50	052 or 120	* or P	
0D	015		undefined	00	000		null
0E	016		undefined	00	000		null
10	020	¯ or 2		5E or 32	136 or 062	¯ or 2	
13	023	+ or χ		25 or 66	045 or 146	+ or χ	
15	025		undefined	00	000		null
16	026		undefined	00	000		null
19	031	⍳ or I		69 or 49	151 or 111	⍳ or I	
1A	032	' or K		27 or 4B	153 or 113	' or K	
1C	034		UCS or UPPERCASE	0E	016		shift out§
1F	037		LCS or LOWERCASE	0F	017		shift in§
20	040	" or 1		23 or 31	042 or 061	" or 1	
23	043	∇ or G		67 or 47	147 or 107	∇ or G	
25	045	⌈ or S		73 or 53	163 or 123	⌈ or S	
26	046	Δ or H		68 or 48	150 or 110	Δ or H	
29	051	⍴ or R		72 or 52	162 or 122	⍴ or R	
2A	052	⌊ or D		64 or 44	144 or 104	⌊ or D	
2C	054		undefined	00	000		null
2F	057		HT or TAB	09	011		horizontal tabulate
31	061	U or V		76 or 56	166 or 126	U or V	
32	062	↓ or U		75 or 55	165 or 125	↓ or U	
34	064	⌄ or 9		21 or 39	041 or 071	⌄ or 9	
37	067	- or +		2D or 2B	055 or 053	- or +	
38	070	≠ or 8		22 or 38	042 or 070	≠ or 8	
3B	073	; or ,		3B or 2C	073 or 054	; or ,	
3D	075		IL or IDLE or NULL	00	000		null
3E	076		PRE or PREFIX	1B	033		escape
40	100	space		20	040	space	
43	103	∘ or J		6A or 4A	156 or 112	∘ or J	
45	105	⍬ or 0		6F or 4F	157 or 117	⍬ or 0	
46	106	□ or L		6C or 4C	154 or 114	□ or L	
49	111	) or]		29 or 5D	051 or 035	) or]	
4A	112	∈ or E		65 or 45	145 or 105	∈ or E	
4C	114		undefined	00	000		null
4F	117		undefined	13	023		null
51	121	: or .		3A or 2E	072 or 056	: or .	
52	122	⍱ or N		6E or 4E	156 or 116	⍱ or N	
54	124	⊂ or Z		7A or 5A	172 or 132	⊂ or Z	
57	127		undefined	00	000		null
58	130	≥ or 6		7C or 36	174 or 066	≥ or 6	
5B	133	? or Q		3F or 51	077 or 121	? or Q	
5D	135		BS or BACKSPACE	08	010		backspace
5E	136		EOB	17	027		end transmission block§
61	141	| or M		6D or 4D	155 or 115	| or M	
62	142	⊃ or X		78 or 58	170 or 130	⊃ or X	
64	144	∧ or 0		26 or 30	045 or 060	∧ or 0	
67	147	↑ or Y		79 or 59	171 or 131	↑ or Y	
68	150	> or 7		3E or 37	076 or 067	> or 7	
6B	153	(or [		28 or 5B	050 or 133	(or [	
6D	155		NL or CR or RETURN	0D	015		carriage return
6E	156		LF or LINE FEED	0A	012		line feed
70	160	< or 3		3C or 33	074 or 063	< or 3	
73	163	_ or F		5F or 46	137 or 106	_ or F	
75	165	⍵ or W		77 or 57	167 or 127	⍵ or W	

TABLE A-23. APL CHARACTER CODE TRANSLATIONS, CORRESPONDENCE CODE CONSOLES IN TERMINAL CLASS 4 (2741) (Contd)

Terminal Correspondence Code				Network ASCII (Normalized Mode Use)			
Hex Code†	Octal Code†	Correspondence Code APL Graphic††	Control Character	Hex Code†††	Octal Code†††	ASCII-APL Graphic	Control Character
76	166	⊥ or B		62 or 42	142 or 102	⊥ or B	
79	171	∝ or A		61 or 41	141 or 101	∝ or A	
7A	172	∩ or C		63 or 43	143 or 103	∩ or C	
7C	174		EOT	04 or 14	004		end of transmission§
7F	177		DEL	18	030		cancel
00	000	space§§		27	047	′	
00	000	space§§		60	140	◇	
00	000	space§§		7B	173	{	
00	000	space§§		7D or 7E	175 or 176	} or ⊢	
3D	075		IL or IDLE or NULL§§	01	001		start of header
3D	075		IL or IDLE or NULL§§	02	002		start of text
3D	075		IL or IDLE or NULL§§	03	003		end of text
3D	075		IL or IDLE or NULL§§	05	005		enquire
3D	075		IL or IDLE or NULL§§	07	007		bell
3D	075		IL or IDLE or NULL§§	0B or 0C	013 or 014		vertical tabulate or form feed
3D	075		IL or IDLE or NULL§§	10	020		data link escape
3D	075		IL or IDLE or NULL§§	12	022		device control 2
3D	075		IL or IDLE or NULL§§	14 thru 16	024 thru 026		device control 4, negative acknowledge, or synchronize
3D	075		IL or IDLE or NULL§§	18 thru 1F	030 thru 037		cancel, end of media, substitute, file separator, group separator, record separator, or unit separator

†Shown with odd parity; odd parity is the default for this terminal class. (Unless PA=N or PA=I, the application program receives the same code as in normalized mode.)

††Each input line is assumed to begin in lowercase. Input characters are translated to lowercase ASCII characters unless prefixed by the UCS code. Once a case shift occurs, it remains in effect until another case shift code is received, the page width is reached, or the line is transmitted to the host computer. During output, case is preserved by insertion of case shift codes where needed.

†††Shown with zero parity (eighth or uppermost bit is always zero).

§Not transmitted to the host computer after translation during input.

§§Output translation only.

TABLE A-24. FULL ASCII NORMALIZED MODE APL CHARACTER SET

128-Character Set

96-Character Subset

64-Character Subset

Bits b_7 b_6 b_5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 ROW / COLUMN	0	1	2	3	4	5	6	7
0 0 0 0 0	NUL 000	DLE 020	SP 040	0 060	≤ 100	P 120	◇ 140	← 160
0 0 0 1 1	SOH 001	DC1 021	⋆⋆ or ∨ 041	1 061	A 101	Q 121	∝ 141	→ 161
0 0 1 0 2	STX 002	DC2 022	≠ 042	2 062	B 102	R 122	⊥ 142	ρ 162
0 0 1 1 3	ETX 003	DC3 023	: 043	3 063	C 103	S 123	∩ 143	⌐ 163
0 1 0 0 4	EOT 004	DC4 024	$ 044	4 064	D 104	T 124	⌊ 144	~ 164
0 1 0 1 5	ENQ 005	NAK 025	÷ 045	5 065	E 105	U 125	∈ 145	↓ 165
0 1 1 0 6	ACK 006	SYN 026	∧ 046	6 066	F 106	V 126	X 146	∪ 166
0 1 1 1 7	BEL 007	ETB 027	′ 047	7 067	G 107	W 127	∇ 147	ω 167
1 0 0 0 8	BS 010	CAN 030	(050	8 070	H 110	X 130	Δ 150	⊃ 170
1 0 0 1 9	HT 011	EM 031	) 051	9 071	I 111	Y 131	⍳ 151	↑ 171
1 0 1 0 A	LF 012	SUB 032	* 052	: 072	J 112	Z 132	∘ 152	⊂ 172
1 0 1 1 B	VT 013	ESC 033	+ 053	; 073	K 113	[133	⊣ 153	{ 173
1 1 0 0 C	FF 104	FS 034	, 054	< 074	L 114	\ 134	⌷ 154	≥ 174
1 1 0 1 D	CR 015	GS 035	- 055	= 075	M 115	] 135	| 155	} 175
1 1 1 0 E	SO 016	RS 036	. 056	> 076	N 116	¯ 136	⊤ 156	⊢ 176
1 1 1 1 F	SI 017	US 037	/ 057	? 077	O 117	_ 137	o 157	DEL† 177

†The graphic 95-character subset does not include DEL; refer to Terminal Transmission Modes in the text.

LEGEND:

Numbers under characters are the octal values for the 7-bit character codes used within the network.

INFORMATIVE AND DIAGNOSTIC MESSAGES B

Informative and diagnostic messages appearing at the terminal are listed alphabetically in table B-1.

Table B-1 contains the following information:

- Message column

  Messages are capitalized; all variables are lowercase.

- Significance column

  Briefly describes the problem and defines variables in the message.

- Action column

  States the action required and how to perform it.

- Issued By column

  States the program that generates the message:

  Network Validation Facility is identified as NVF.

  Communications Control Program is identified as CCP.

  Terminal Interface Program is identified as TIP.

  Terminal Verification Facility is identified as TVF.

TABLE B-1. INFORMATIVE AND DIAGNOSTIC MESSAGES

Message	Significance	Action	Issued By
APPLICATION BUSY, TRY AGAIN LATER. terminalname - APPLICATION:	Another terminal user with the same user name index is currently logged into the chosen application.	Enter the name of another application, or try again later.	NVF
APPLICATION FAILED. aname CONNECT TIME hh.mm.ss terminalname - APPLICATION:	The application has failed. The parameter aname is the name of the failed application. The parameter hh:mm:ss specifies the length of time device terminalname was connected to the application.	Enter the name of another application program.	NVF
APPLICATION NOT PRESENT terminalname - APPLICATION:	The requested application program is not executing at a control point; terminalname represents the device name.	Enter the name of another application, or try again later.	NVF
APPLICATION RETRY LIMIT	The permitted number of unsuccessful attempts to enter a legal application name have been made.	Hang up and get help.	NVF
AUTO END - TVF TERMINATES	The network software has sent notification of a pending shutdown to TVF. All device connections are automatically ended. No more TVF commands can be entered.	Contact host operator and ask when TVF will again be available.	TVF
CONNECTION REJECTED. terminalname - APPLICATION:	The requested application program has refused connection with the device terminalname.	The action to take depends on the application involved. Enter the name of another application, or try again later.	NVF
DEL	Cancel-input character has been processed.	None.	TIP

TABLE B-1. INFORMATIVE AND DIAGNOSTIC MESSAGES (Contd)

Message	Significance	Action	Issued By
ERROR - ENTRY entry	TVF cannot interpret the type-in identified as entry; entry consists of the first 5 characters of the device's last processed transmission. TVF processing stops.	To resume testing, enter a command. Check the command for a typographical error before transmitting it.	TVF
ERROR - PARITY - entry	TVF received a parity error indication with the type-in identified as entry; entry consists of the first 5 characters of the device's last processed transmission. TVF processing stops.	To resume testing, enter a command.	TVF
ERROR - TVF READY	TVF has recovered from an internal software problem. TVF processing has stopped.	To resume testing, enter a command. If this message occurs repeatedly, notify site maintenance personnel.	TVF
FAMILY:	Normal host prompt for family name.	Enter a valid NOS family name.	NVF
FROM NOP...string	A message from the network operator is displayed as string.	None.	CS
HOST AVAILABLE	You have a selected and available host system, to which your terminal can be connected.	As indicated by the prompt message: ENTER INPUT TO CONNECT TO HOST	CCP
HOST BUSY	The requested connection was rejected by the host. Not enough resources are currently available to service another terminal.	As indicated by the prompt message: ENTER HD TO SEE HOST STATUS, wait a short time and try again, or try another host.	CCP
HOST CONNECTED	Your terminal is connected to a host.	As indicated by the prompt message: READY FOR INPUT or TERMINAL DISABLED BY NOP.	CCP
HOST DISCONNECTED	The NPU has ended your device's connection with the host; connection was not completed or was broken.	As indicated by the prompt message: ENTER INPUT TO CONNECT TO HOST or TERMINAL DISABLED BY NOP.	CCP
HOST UNAVAILABLE	You have selected and attempted to establish a connection with a host that is currently not communicating with the network through a path you can use.	As indicated by the prompt message: ENTER HD TO SEE HOST STATUS.	CCP
ILLEGAL APPLICATION, TRY AGAIN terminalname - APPLICATION:	The name used to request an application program is not recognized or is not valid for the user name.	Correct the entry, or enter the name of another application program. If the entry was correct, contact a site administrator.	NVF
IMPROPER LOGIN, TRY AGAIN.	An unacceptable family name, user name, or password has been entered.	Restart the login procedure by entering a valid family name or user name, as requested by the prompt following this message.	NVF
INPUT DISCARDED..	CCP has discarded part of your last input transmission because of a temporary storage problem.	See Suspensions of Communication in section 4.	CCP

TABLE B-1. INFORMATIVE AND DIAGNOSTIC MESSAGES (Contd)

Message	Significance	Action	Issued By
INPUT IGNORED	NVF has discarded part of your last input transmission because either a user break occurred or the application program that the data was intended for has failed.	None.	NVF
LOGGED OUT.	Terminal has been queued for disconnection from the host.	Hang up phone or initiate new host connection procedure.	NVF
LOGIN TERMINATED.	You cannot be validated as a user of this host. Your terminal will be disconnected.	Try another host.	NVF
NO HOST AVAILABLE	No hosts are currently communicating with the network through a path you can use.	As indicated by the prompt message: ENTER HD TO SEE HOST STATUS or try again later.	CCP
NO HOST CONNECTED	You are not connected to a host and/or you entered the TM command.	As indicated by the prompt message: ENTER INPUT TO CONNECT TO HOST.	CCP
NO HOST SELECTED	A host is available, but you have not selected one with which to communicate.	As indicated by the prompt message: ENTER HN=NN TO SELECT HOST.	CCP
OVER..	The end of a page of output data has been reached but more output is available.	Enter an empty line to receive more data.	TIP
PASSWORD:	Normal host prompt for password.	Enter a valid password.	NVF
REPEAT..	The temporary storage space problem has ended.	Reenter your previous input.	TIP
SYSTEM CLOSED	Host shutdown is in progress and login is not permitted.	Try again later.	NVF
TC INCORRECT TERMINAL CLASS	The value you specified in a TC command is not within the range allowed for your current terminal class.	Check for a typographical error.	TIP
TIMEOUT	The most recent prompting message was not responded to within 2 minutes. Terminal disconnect from the host was requested.	If the terminal disconnects, a new host connection must be established before login can begin.	NVF
TVF TEST COMPLETE	Self-explanatory.	Check test results and respond to any subsequent prompting line.	TVF
USER ACCESS NOT POSSIBLE - CONTACT NETWORK ADMIN.	The user is not validated by NOS to access any application programs, the user name security count has been exceeded, a login mandatory application program name is not recognized by NVF, or two different mandatory application programs are defined for your user name.	Contact host administrative personnel.	NVF
USER NAME:	Normal host prompt during login for user name.	Enter the assigned user name.	NVF
USER RETRY LIMIT.	Four unsuccessful login attempts have been made.	Contact host administration to determine user validation requirements.	NVF

TABLE B-1. INFORMATIVE AND DIAGNOSTIC MESSAGES (Contd)

Message	Significance	Action	Issued By
VOLUNTEERED APPLICATION IGNORED.	A site-defined mandatory value overrode the value you supplied.	None.	NVF
VOLUNTEERED USER/PASSWORD IGNORED.	A site-defined mandatory value overrode the value you supplied.	None.	NVF
WAIT..	Because of a temporary overload condition, the last transmitted line from your terminal's console was discarded.	Wait for the REPEAT message, then repeat input.	TIP
image > BAD /*CONFIG.	The /*CONFIG statement cannot be processed past the characters shown as image.	Correct the syntax or value error and resend the statement.	TIP
image, BAD /*CONFIG.	The statement indicated by image was received when a /*CONFIG statement was expected.	Enter a valid /*CONFIG statement.	TIP
application CONNECT TIME hh.mm.ss terminalname - APPLICATION:	Control of the terminal has been returned to NVF from an application. The parameter application is the name of the application previously selected. The parameters hh:mm:ss indicate the length of time device terminalname was connected to the application.	Enter the name of another application program.	NVF
terminalname - APPLICATION:	Prompt for application program; terminalname represents the device name.	Enter name of desired application program.	NVF
..	Prompt for entry of another command.	Enter another command.	TVF
xx ACCEPTED..	Informative message that indicates a valid terminal definition command was entered; xx is the mnemonic of the command parameter.	None.	TIP
xx DUPLICATE CHARACTER	The value you specified in the terminal definition command xx has already been used for another network function character.	Try another value.	TIP
xx INCORRECT COMMAND	The characters xx are not a terminal definition command mnemonic or a priority data character command.	Check for a typographical error.	TIP
xx INCORRECT VALUE	You specified an invalid value in the terminal definition command xx.	Check for a typographical error. Check the range of values described in sections 2, 4, or 5 for the command.	TIP
xx VALUE INAPPROPRIATE	You specified a value in the terminal definition command xx that is not appropriate for devices in your device's current terminal class.	Check for a typographical error. Check the range of values described in sections 2, 4, or 5 for the command.	TIP

GLOSSARY C

This glossary defines terms and mnemonics unique to the description of the software presented in this manual. It also contains terms whose interpretation within this manual is intended to be more constrained or different from that commonly made. This glossary supplements the American National Dictionary for Information Processing (ANSI X3/TR-1-77, FIPS 11-1). Some terms used in other documentation for the network software are included for the reader's convenience when reconciling terminology.

Acknowledgment, Block -
A message returned to the sender confirming the delivery of one block; referred to as BACK in CCP documentation.

Address -
A location of data (as in the main or micro NPU memory) or of a device (as a peripheral device or terminal).

APL -
A scientific programming language characterized by powerful operators and special graphic symbols.

Application Block Header (ABH) -
A single 60-bit word description accompanying every block passing between an application program and NAM.

Application Block Limit (ABL) -
The number of unacknowledged blocks a logical connection is allowed to have outstanding (queued by the network) at any one time.

Application Block Number (ABN) -
A field in the application block header. An application-assigned number used to identify a particular network data block.

Application Block Type (ABT) -
A field in the application block header defining the accompanying block as either data or supervisory, null or not null, and indicating whether this is the last block of a message.

Application Interface Program (AIP) -
A group of routines that reside in the application program's field length. These routines translate and buffer communication between the application program and the network, using the system control point feature of NOS.

Application Name (aname) -
Up to seven 6-bit letters or digits (the first must be a letter) used to identify an application program. It is used by another application program or by a terminal operator when connection to the application is requested, and by the host operator when issuing commands to the program.

Application Program -
A program resident in a host computer that provides an information storage, retrieval, and/or processing service via the data communication network and the Network Access Method. Application programs always use the system control point feature of NOS to communicate with the Network Access Method. In the context of network software, an application program is not an interactive job, but rather a terminal servicing facility. A terminal servicing facility provides terminal users with a specific processing capability such as remote job entry from batch terminals, transaction processing, entry and execution of interactive jobs, and so forth. For example, the standard CDC interactive facility IAF makes terminal input and output appear the same to an executing program as file input and output; IAF is a network application program, but the executing program using IAF is an interactive job.

Archetype Terminal -
The specific terminal equipment possessing all of the attributes used as defaults for the definition of one terminal class. Each terminal class has a corresponding archetype terminal.

Asynchronous -
A transmission in which each information character is individually synchronized by the use of start and stop bits. The gap between each character is not necessarily of fixed length.

Asynchronous Protocol -
The protocol used by asynchronous, teletypewriter-like devices. For CCP, the protocol is actually the set of protocols for eight types of real terminals. The NPU/terminal interface is handled by the ASYNC TIP.

Automatic Input -
A feature that permits data sent to a terminal at the end of an output message to be returned to the host application program as the beginning of the next input from the terminal. The most common use of automatic input is to send line numbers to the terminal console for return as part of the entry for a text editor or file generator program.

Automatic Login -
The process whereby one or more of the Network Validation Facility login dialog parameters is supplied to NVF from the local configuration file. Parameters supplied through automatic login configuration of a terminal device suppress prompting for the corresponding dialog entries and override any entries made from the terminal console.

Automatic Recognition -
The process whereby the Terminal Interface Program identifies characteristics of a terminal when the terminal's communication line becomes active. The Terminal Interface Program determines sub-TIP type subset of protocol type) and line speed (and, for mode 4 terminals, the cluster and terminal addresses) by various methods for lines configured for automatic recognition. The Communications Supervisor then matches these parameters against the descriptions of specific terminals in the network configuration file; the terminal with the closest match to the empirically determined parameters is automatically recognized as the terminal on the communication line.

Base System Software -
The set of programs in CCP that supplies the monitor, timing, interrupt handling, and multiplexing functions for the NPU. Base system software also includes common areas, diagnostics, and debugging utilities.

Batch Device -
A device that is capable of conducting input only or output only operations. Card readers, line printers, and plotters are examples of batch devices. Batch devices are sometimes referred to as passive devices. A batch device is an input-only or output-only device or portion of a terminal, and is usually serviced by the Remote Batch Facility program using the PRU interface.

Batch Mode -
The state of a mode 4A terminal during which batch data is transmitted from the terminal's card reader to central files and/or from central files to the terminal's line printer. Also, the state of an asynchronous terminal during which input transmission occurs on a block-by-block basis.

Binary Synchronous Communications (BSC) -
A communications protocol supported by the BSC TIP. This protocol connects IBM 2780 or 3780 terminals to the NPU using half-duplex synchronous transmissions in a point-to-point mode. The terminals have batch devices that use EBCDIC code. Transparent data exchanges are permitted. The terminals are configured to have a virtual console (interactive device). This is composed of a card reader for input and a printer for output.

Block -
In the context of network communications, a portion or all of a message. A message is divided into blocks to facilitate buffering, transmission, error detection, and correction for variable length data streams. Differing block protocols apply to the host/NPU and the NPU/terminal interfaces.

During input from a terminal, a block is a single transmission comprising one or more lines of one or more messages.

During input to an application program, a block is a single line comprising part or all of a message. Terminal transmission blocks are divided into as many application program input blocks as needed, until the message is completed.

During output from a host application program, a block is one or more lines. During output from an NPU to a terminal, a block is one terminal transmission buffer.

Block Acknowledgment -
See Acknowledgment, Block.

Block Header -
See Application Block Header.

Block Limit -
The number of message blocks that can be awaiting delivery at any one time in either the host-to-NPU direction or the NPU-to-host direction for a single terminal.

Block Type -
See Application Block Type.

Break -
A method employed by a terminal operator to interrupt output or input in progress. See also, Interrupt.

Buffering -
The process of collecting data together in buffers. Ordinarily, no action on the data is taken until the buffer is filled. Filled buffers include the case where data is terminated before the end of the buffer and the remaining space is filled with irrelevant codes.

Byte -
A group of contiguous bits. Unless prefixed (for example, a 6-bit byte), the term implies 8-bit groups. When used for encoding character data, a byte represents a single character.

Cassette -
The magnetic tape device in an NPU used for bootstrap loading of off-line diagnostics and (in remote NPUs) the bootstrap load/dump operation.

CE Error Message -
A message containing information concerning hardware and/or software malfunctions.

Character -
Any alphabetic, numeric, or special symbol that can be encoded. This term applies to the graphic characters for a terminal input or output device, and to the encoded control characters used by the terminal. Within the network software, a character is a coded byte of data, such as a 6-bit display code or 7-bit ASCII code. Terminals use a wide range of codes to represent a specific character. Network products translate between terminal codes and host codes. Unless otherwise specified, references to characters in this manual are to ASCII 7-bit byte characters.

Character Mode -
Refers to terminal transmission size, when each character is transmitted to the network processing unit as it is entered at the terminal.

Character Type -
See Application Character Type.

Circular Input Buffer (CIB) -
Within CCP, a buffer used by the multiplex subsystem to collect all data passing upline from the multiplexer.

Cluster -
Mode 4 devices grouped into a terminal by a common cluster address. Synonymous with terminal.

Cluster Address -
The hardware address of a cluster. This term is used in several ways within mode 4 documentation, as shown in table C-1.

TABLE C-1. MODE 4 NOMENCLATURE EQUIVALENCE

Networks Nomenclature	Mode 4A Nomenclature	Mode 4C Nomenclature
Network processing unit	Data source	Control station
Cluster address	Site address	Station address
Cluster controller	Equipment controller	Station
Terminal address	Station address	Device address
Terminal	Equipment controller	Station
Device	Equipment	Device

Command Driver -
The hardware driver that controls the multiplex subsystem.

Communication Element -
Any entity that constitutes a point of input to, or output from, the data communication network. This includes terminal devices, terminals, and communication lines.

Communication Line -
A complete communication circuit between a terminal and its network processing unit.

Communication Network -
The portion of the total network comprising the linked network processing units. The communication network excludes host computers and terminals.

Communications Control Program (CCP) -
A portion of the network software that resides in a 255x Series network processing unit. This set of modules performs the tasks delegated to the NPU in the network. This software can include such routines as the Terminal Interface Program.

Communications Supervisor (CS) -
A portion of the network host software, written as an application program; the Communications Supervisor configures and controls the status of network processing units and all of their communication lines and terminals.

Configuration -
See Network Configuration.

Connection Number (CN) -
A unique number assigned to each active device on a logical link.

Constant Carrier -
A communication line with a transmission carrier signal that remains on continuously; failure is reported if the carrier signal received remains off for a period of time that equals or exceeds a failure verification period.

Contention -
The state that exists in a bidirectional transmission line when both ends of the line try to use the line for transmission at the same time. All protocols contain logic to resolve the contention situation.

Control Blocks -
(1) The types of blocks used to transmit control (as opposed to data) information; (2) Blocks assigned for special configuration/status purposes in the NPU. The major blocks are line control blocks (LCB), logical link control blocks (LLCB), logical channel control blocks (LCCB), terminal control blocks (TCB), queue control blocks (QCB), buffer maintenance control blocks (BCB), multiplexer line control blocks (MLCB), text processing control blocks (TPCB), and diagnostics control blocks (DCB).

Controlled Carrier -
A communication line with a transmission carrier signal that is raised and lowered with each block transmitted; failure is reported if the carrier signal received does not fluctuate in a similar fashion.

Controlled Terminal -
A terminal whose input can be started and stopped by the network software. When a terminal places data on a communication line only in response to a poll, the maximum input rate can be controlled by controlling the polling rate. Mode 4 terminals are controlled.

Coupler -
A hardware module resident in a front-end network processing unit. That coupler links the network processing unit to a host computer.

Cross -
The software support system for CCP. These programs, which are run on the host, support source code programming in Pascal, macroassembler, and microassembler languages. The compiled or assembled output of the Cross programs are in object code format on host computer files. The object code files are processed by other Cross programs and host installation programs into a downline load file for an NPU.

Cyclic Redundancy Check (CRC) -
A check code transmitted with blocks/frames of data. It is used by several protocols.

Data -
Any portion of a message created by the source, exclusive of any information used to accomplish transmission of such a message.

Data Buffer -
A block of 64 contiguous words in CCP used for storing data (two characters per word). A buffer usually has a header of one or more words. Data within a data buffer is delimited by pointers to the first and last characters (data buffers are character-oriented). If the data cannot all fit into one buffer, an additional buffer is assigned and is chained to the current buffer. Buffer assignment continues until an entire data block is contained in the chain of buffers. Buffers are chained together only for upline data.

Data Compression -
The technique of transmitting a sequence of identical characters as a control character and a number representing the length of the sequence.

Data Set -
A hardware interface that transforms analog to digital data and the converse. A data set is a modem capable of using telephone lines.

Dedicated Line -
A communication line that is permanently connected between a terminal and a network processing unit. Contrast with Switched Line.

Default -
A host-supplied or network-supplied parameter value or name used when a value or name is not supplied by the user.

DEFINE -
An NDL statement that provides the macro-like capability of substituting an identifier in coding for a more complex entity. When the coding is processed, the identifier is interpreted as if it had been replaced by the complex entity. Also, a NOS command that creates permanent files.

Destination -
The terminal or application program designated to receive the message.

Destination Node (DN) -
The NPU node that directly interfaces to the destination of a data block. For instance, the destination node of an upline block might be the network host software that passes the block to the application program responsible for processing the block.

Device -
A separately addressable portion or all of a terminal. This term is used in various ways within mode 4 documentation, as shown in table C-1.

Diagnostics -
Software programs or combinations of programs or tables that aid the troubleshooter in isolating problems.

Direct Access File -
In the context of NOS permanent files, a file that is accessed and modified directly.

Display Code -
A character code used internally in a CDC CYBER 170 host computer. Each character consists of six bits (two octal digits).

Downline -
The direction of output information flow, from a host computer application program.

Dump -
In the context of CCP, the process of transferring the contents of the NPU main memory, registers, and file 1 registers to the host. The dump can be processed by the Network Dump Analyzer in the host to produce a listing of the dumped information.

Echo -
The process of displaying a keystroke on a console. Echoing can be done from the TIP, from a modem, or from the terminal itself.

Echoplex -
The process of returning received characters on a full-duplex line. Not all terminals on full-duplex communication lines are capable of echoplex operation.

File -
A unit of batch data. Files are transferred between application programs and terminals by using PRUBs on the NPU's host side and transmission blocks on the NPU's terminal side. A file contains one or more records. Example: a card reader job consists of a file containing the card image records of all the cards in the job deck.

File Registers -
The two sets of microregisters (file 1 and file 2) in a 255x series network processing unit. File 1 registers contain parameter information that is reloaded whenever the NPU is initialized. Microprograms using file 1 registers can also change values in them. File 2 registers are invariant firmware registers that come preprogrammed with the network processing unit.

Format Effectors (FE) -
Characters in an output data stream that determine the appearance of data at the console. A format effector usually takes the form of a single character in the output line. For printing devices, the character is translated by the output side of the TIP into a combination of carriage returns, line feeds, or spaces. Similarly, FEs for displays can command new lines, screen clearing, or cursor positioning.

Frame -
A frame is a block of data sent across a high-speed link. It is composed of control bytes, a CRC sum, and (in some cases) data bytes in sub-block sequence. A sub-block can be a network data block or a part of a block. The frame is the basic communication unit used in trunk (NPU to NPU) communications and provides high-data density in bit-serial format over data-grade lines, as well as data assurance.

Frames are transmitted as a sequence of bytes through the multiplex subsystem that uses a hardware-controlled frame on the input and output multiplex loops.

Free-Wheeling Terminal -
A terminal that can input at the discretion of the terminal user and has an input rate that cannot be controlled directly. Asynchronous terminals are free-wheeling. Contrast with Controlled Terminal.

Front-End NPU -
A network processing unit that directly interfaces to one or more hosts. Synonymous with local NPU.

Full Duplex (FDX) -
Two-way simultaneous transmission on a communication line.

Function Codes -
Codes used by the CCP service module to designate the type of function (command or status) being transmitted. Two codes are defined: primary function code (PFC) and secondary function code (SFC). Function codes are also used between NAM and the host application programs in all supervisory messages.

Half Duplex (HDX) -
Two-way alternating transmission on a communication line. Normally a single set of data lines carry input, output, and part of the control information. Contention for use is possible in half-duplex mode and must be resolved by the protocol governing line transfers.

Halt Codes -
Codes generated by the NPU when it is stopped by its software. These codes, which indicate the cause of the stoppage, are contained in a CCP dump.

HASP -
A protocol based on the BSC protocol; it is used by Houston Automatic Spooling Program (HASP) workstations. A workstation has both interactive and batch devices. The standard code of all HASP devices is EBCDIC; however, transparent batch data exchanges with the host are also permitted. The HASP TIP converts interactive HASP data from EBCDIC transmission blocks to ASCII network data blocks; it converts batch HASP data from EBCDIC transmission blocks to PRU blocks; nontransparent batch data is converted to display code.

Header -
The portion or portions of a block holding information about the block source, destination, and type. During network movement, a block can acquire several headers. For example, during movement of a block from a terminal to the host over an X.25 network, the block acquires the following headers: one at the terminal (also a trailer), one for the frame, one for the packet, and another for the host application program. Headers are discarded by the appropriate stage of processing, so that in this example, the host sees only the application program block header. Conversely, headers are generated and discarded as needed downline, so that the terminal sees only the terminal header (and trailer).

High-Speed Synchronous Line -
A data transmission line operating at or above 19200 b/s. These lines are normally used for local LIP/remote LIP transfers and for X.25 and HASP network transfers.

Host -
A computer that executes network application programs and can access the network.

Host Interface Package (HIP) -
The CCP program that handles block transfers across the host/local NPU interface. The HIP transfers control blocks and network data blocks.

Host Node -
The node ID number of the NPU coupler that directly interfaces with a host computer.

Host Operator (HOP) -
The operator who resides at the system console, initiates NAM, and controls NPUs and network-related host elements. The HOP may do all network operator functions as well as those functions unique to the HOP despite the existence of NPU operators. There can be only one HOP for each host. Contrast with Network Operator.

Initialization -
The process of loading an NPU and optionally dumping the NPU contents. After downline loading from the host, the NPU network-oriented tables are configured by the host so that all network processors have the same IDs for all network terminals, lines, trunks, etc.

Input -
Information flowing upline from terminal to host computer.

Input Buffer -
In the context of CCP, a data buffer containing part of a network data block destined for a host application program. These buffers are dynamically allocated and released.

Interactive Device -
Any device capable of conducting both input and output, making it capable of dialog with the Network Validation Facility. Also known as a console device. Contrast with Passive Device. An interactive device is serviced by an application program using the interactive virtual terminal interface.

Interactive Facility (IAF) -
The network host product application that supports interactive (conversational) job execution and command processing. During interactive job processing, files can be created and edited from a terminal.

Interactive Mode -
The state of a mode 4A terminal during which keyboard data (such as user commands) is transmitted to the host computer and/or messages are transmitted to the terminal's console.

Interactive Virtual Terminal Interface -
The set of protocols and characteristics common to all logical terminal devices serviced by the network software as interactive devices with combined input and output units.

Interrupt -
An event in network operation that is sometimes synonymous with a break condition. An interrupt causes a pause or termination in an operation being performed by the terminal, terminal Interface Program, or network application program. Terminals can have one or more keys that cause distinctly different interrupts, as described in section 2.

Level -
For logical records, an octal number 0 through 17 in the system-supplied 48-bit marker that terminates a short or zero-length PRU.

Line -
(1) During input, the portion of a transmission that precedes a logical end-of-line indicator or a physical end-of-line delimiter, such as a line feed. A line can contain part or all of a message.

During output, the portion of a message that is output on a single physical line of the terminal output device.

(2) A hardware connection between an NPU and a terminal or group of terminals.

Link -
A connection between two NPUs or an NPU and a host.

Link Interface Package (LIP) -
The CCP program that handles frame transfers across a trunk; that is, across the link connection between a local and a remote NPU. A LIP uses CDCCP protocol and interfaces on the local NPU side to the HIP. On the remote NPU side, the LIP interfaces with the appropriate TIP. In both local and remote NPUs, the LIP interfaces with the multiplexer subsystem for transfer across the trunk.

Load -
The process of moving programs downline from the host and storing them in the NPU macromemory and micromemory. Loading of a remote NPU is accomplished by the host through the use of the LIP in the local NPU.

Local Configuration File (LCF) -
A file in the host computer system, containing information on the logical relationships among the service elements in the network. The file contains a list of the application programs available for execution in the host computer, and the users that require automatic login to them. This is a NOS direct access permanent file.

Local NPU -
An NPU that is connected to the host via a coupler. A local NPU always contains a HIP for processing block protocol transfers across the host/local NPU interface. Synonymous with front-end NPU. Contrast with Remote NPU.

Logical Connection -
A logical message path established between two application programs or between a network terminal device and an application program. Until terminated, the logical connection allows messages to pass between the two entities.

Logical Line -
The basic message unit of a console device.

During input, all character codes between two occurrences of the character code associated with the transmission key for the terminal (usually, the carriage return or ETX keys).

During output, all character codes between two occurrences of a logical line terminator byte.

Contrast with Physical Line.

Logical Link (LL) -
The portion of a logical connection defined by host node and terminal node ID numbers. A logical link is an error-free path across the network over which many separate logical connections are multiplexed. A logical link cannot traverse more than two NPUs.

Logical Record -
Under NOS, a data grouping that consists of one or more PRUs terminated by a short PRU or zero-length PRU. Equivalent to a system-logical-record under NOS/BE.

Logical Terminal -
The terminal the network software believes it is communicating with. Compare with Physical Terminal. Interactive logical terminal devices (consoles) are distinct from passive logical terminal devices (batch devices).

Loop Multiplexer (LM) -
The hardware that interfaces the CLAs (which convert data between bit-serial digital and bit-parallel digital character format) and the input and output loops.

Low/Medimum-Speed Voice-Grade Line -
A line that operates at bit transmission rates at or below 19200 bits per second. These lines usually connect individual terminals to an NPU or to an X.25 PAD service.

Macromemory -
The portion of 255x series network processing unit memory that contains code involved in data communication, such as the Terminal Interface Program. The macromemory is partly dedicated to programs and common areas; the remainder is buffer area used for data and overlay programs. Word size is 16 data bits plus three additional bits for parity and program protection. Memory is packaged in 16K and 32K word increments.

Message -
A logical unit of information, as processed by an application program. When transmitted over a network, a message can consist of one or more blocks.

During input, a message is one or more lines comprising a single command, transaction, or entry. Output cannot occur until transmission of an input message is completed.

During output, a message is all lines of application program transmissions until application program input is expected.

Micromemory -
The portion of 255x series network processing unit memory that contains code defining the unit to itself. This consists of 8192 words of 64-bit length. 1024 words are Read Only Memory (ROM); the remaining words are Random Access Memory (RAM) and are alterable. The ROM memory contains the emulator microprogram that allows use of assembly language.

Microprocessor -
The portion of the NPU that processes the programs.

Mode 4 -
A communication line transmission protocol that requires the polling of sources for input to the data communication network. Control Data defines two types of mode 4 equipment, mode 4A and mode 4C. Mode 4A equipment is polled through the hardware address of the console device, regardless of how many devices interface to the network. Mode 4C equipment is polled through separate hardware addresses, depending on the point each device uses to interface with the network.

Modem -
A hardware device for converting analog levels to digital signals and the converse. Telephone lines interface to digital equipment via modems. Modem is synonymous with data set.

Monitor -
The portion of the CCP base system software responsible for time and space allocation within the computer. The principal monitor program is OPSMON. OPSMON executes OPS level programs by scanning a table of programs that have pending tasks.

Multiplex Loop Interface Adapter (MLIA) -
The hardware portion of the 255x series NPU multiplex subsystem that controls the multiplexing loops (input and output) as well as the interface between the NPU and the multiplex subsystem.

Multiplex Subsystem -
The portion of the base CCP software that performs multiplexing tasks for upline and downline data, and also demultiplexes upline data from the circular input buffer and places the data in line-oriented input data buffers.

Neighbor NPUs -
Two NPUs connected to one another by means of a trunk (link).

Network -
An interconnected set of network processing units, hosts, and terminal devices.

Network Access Method (NAM) -
A software package that provides a generalized method of using a communication network for switching, buffering, queuing, and transmitting data. NAM is a set of interface routines used by a terminal servicing facility for shared access to a network of terminals and other application programs, so that the facility program does not need to support the physical structures and protocols of a private communication network.

Network Address -
The address used by block protocol to establish routing for the message. It consists of three parts: DN, the destination node; SN, the source node; and CN, the connection number.

Network Configuration -
The process of setting tables and variables throughout the network to assign lines, links, terminals, and so forth, such that all elements of the network recognize a uniform addressing scheme. After configuration, network elements accept all data commands directed to/through themselves and reject all other data and commands.

Network Configuration File (NCF) -
A network definition file in the host computer, containing information on the network and communication elements and permissible linkages between them. The status of the elements described in this file is modified by the network operator in the course of managing the network through the Communications Supervisor. This is a NOS direct access permanent file.

Network Definition File -
Either of the two types of NDL program output files that determine the configuration of the network. This can be a network configuration file or a local configuration file.

Network Definition Language (NDL) -
The compiler-level language used to define the network configuration file and local configuration file contents.

Network Definition Language Processor (NDL Processor) -
The network software module that processes an NDL program to create the network definition files and other NDL program output.

Network Element -
Any entity in the total network that is not a communication element; this term is usually applied to the data communication network entities comprising the NPUs and their linkages.

Network Operator -
The administrative operator who manages the hardware, linkages, and other network elements of the data communication network by communicating with the Communications Supervisor in a host computer. Contrast with Host Operator. The network operator can also be a host operator.

Network Processing Unit (NPU) -
The collection of hardware and software that switches, buffers, and transmits data between terminals and host computers.

Network Supervisor (NS) -
A portion of the network software, written as a NAM application program. The Network Supervisor dumps and loads the NPUs in the communication network.

Network Validation Facility (NVF) -
A portion of the network software, written as a NAM application program. The Network Validation Facility performs all device login and application program access validation processing and supports login dialog with the terminal user.

Node -
A hardware or software entity that creates, absorbs, switches, and/or buffers network data blocks. NPUs and host couplers are nodes of the CDC network.

Normalized Mode -
This term refers to network transmission type, when terminal character codes are converted by the network processing unit to (and from) ASCII character codes before transmission to the host computer (or after transmission from it). In network normalized transmission mode, input data is completely or partically edited by the Terminal Interface Program to imitate input from terminals that edit input before transmission. Contrast with Transparent Mode.

NPU Operator -
(1) The network operator who uses a terminal and controls network elements such as NPUs, trunks, logical links, lines, and terminals. Contrast with Host Operator.

(2) The diagnostic operator who uses a terminal and tests trunks and communication lines. Contrast with Network Operator.

(3) An operator using the offnet NPU console.

Off-Line Diagnostics -
Optional diagnostics for the 255x series network processing unit that require the NPU to be disconnected from the network.

On-Line Diagnostics -
Optional diagnostics for the 255x series network processing unit that can be executed while the NPU is operating as a part of the network. Individual lines being tested must, however, be disconnected from the network. These diagnostics are provided if the user purchases a network maintenance contract.

OPS Monitor -
The CCP monitor. See Monitor.

Output -
Information flowing from a host application program.

Output Buffer -
Any buffer that is used to hold a downline message from the host.

Owning Console -
The interactive terminal device associated with a given passive terminal device. The owning console enters commands that control the operation of the passive terminal device.

Packet -
A group of binary digits, including data and call control signals, that is transmitted as a single unit. The data, control signals, and error-control information are arranged in a specific format.

Packet Assembly/Disassembly Service (PAD) -
A definition of the procedures for the operation of an asynchronous terminal through a packet-switching network (PSN).

Packet-Switching Network (PSN) -
A type of network that supports data communication between various terminals and computers. The PSN is usually licensed as a common carrier. Terminal interface to a PSN is defined by the packet assembly/disassembly (PAD) service. PSN interface with the NOS network is defined by the X.25 transmission protocol.

PAD SubTIP -
A subTIP of the X.25 TIP that allows asynchronous ASCII terminals to communicate over a packet-switching network.

Paging (Screen) -
The process of filling a CRT display with data and holding additional data for subsequent displays. Changing the paged display is controlled by the terminal operator if the page wait option is selected.

Parity -
A type of data assurance. The most common parity is character parity; that is, the supplying of one extra bit per character so that the sum of all the bits in the character (including the parity bit) is always an even (even parity) or odd (odd parity) number.

Pascal -
A high level programming language used for CCP programs. Almost all CCP programs are written in the Pascal language.

Passive Device -
Any terminal device incapable of conducting both input and output and therefore incapable of dialog with the Network Validation Facility. Batch unit record peripherals are typical examples of passive terminal devices. Also known as a nonconsole device. Contrast with Interactive Device.

Password -
A parameter in the terminal operator's login procedure type-in, used for additional access security by the Network Validation Facility.

Peripheral Processor Unit (PPU) -
The hardware unit within the host computer that performs physical input and output through the computer's data channels.

Physical Line -
A string of data that is determined by the terminal's physical characteristics (page width or line feed). Contrast with Logical Line, which is determined by a carriage return or other message-forwarding signal.

During input, all character codes entered between the occurrence of the last output character, carriage return, or line feed, and the occurrence of the page width boundary, a line feed, or a carriage return code comprise a physical line. During output, all characters between the leftmost and rightmost character positions possible on one line of the output device comprise a physical line.

Physical Link -
A connection between two major network nodes such as neighboring nodes. Messages can be transmitted over active physical links.

Physical Record Unit (PRU) -
Under NOS, the amount of information transmitted by a single physical operation of a specified device. The size of a PRU depends on the device, as shown in table C-2.

A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

TABLE C-2. PRU SIZE

Device	Size in Number of 60-Bit Words
Mass storage	64
Tape in SI format with binary data	512
Tape in I format	512
Tape in other format	Undefined

Physical Terminal -
The set of input and output devices through which the terminal user communicates with the network software. The set consists of only those devices that cannot be separately addressed. See also Terminal.

Polling -
The process of requesting input from hardware or software that only provides input on request. Polling is a concept of several network protocols and is used to avoid input contention. Mode 4 terminals are polled for input by the Terminal Interface Program servicing them; an application program polls all logical connections for input, whether the logical connections are with controlled mode 4 terminals or free-wheeling asynchronous terminals.

Port -
The physical connection in the NPU through which data is transferred to/from the NPU. Each port is numbered and supports a single line. Subports are possible but not used in the current version of CCP.

Primary Function Code (PFC) -
See Function Codes.

Priority -
The condition when traffic through the network is maintained preferentially for one or more devices out of all devices producing network traffic. Terminals with priority are the last devices for which network traffic is suspended when traffic must be temporarily stopped because the network is operating at capacity. Devices with priority receive preferential treatment of their input or output.

Protocol -
A set of standardized conventions that must be used to achieve complete communication between elements in a network. A protocol can be a set of predefined coding sequences, such as the control byte envelopes added to or removed from data exchanged with a terminal; a set of data addressing and division methods, such as the block mechanism used between an application program and the Network Access Method; or a set of procedures used to control communication, such as the supervisory message sequences used between an application program and the Network Access Method.

PRU Block (PRUB) -
Physical record unit block. A block format for batch devices that is compatible with the host's PRU (batch file) handling capabilities. CCP TIPs convert all upline batch data to this format (exception: no transformations are made to transparent data except to put the messages into PRUBs). By this method, application programs in the host need only to be able to process batch data in PRU format rather than in the multiplicity of formats that real terminals use. Downline messages from the host to real batch devices are converted from PRUB to real terminal format. PRUB processing is controlled by the TIPs with the help of the BIP.

PRU Commands -
A set of commands from the host or a terminal that changes batch device or batch file characteristics, alters batch data stream flow, or transmits accounting data to the host. All batch file and batch device commands can come from the host. A few batch file commands can also come from the terminal. Some batch stream flow commands come from the host; others come from the terminals. Accounting data commands come only from the terminals.

PRU Device -
Under NOS, a mass storage device or a tape in SI or I format, so called because records on these devices are written in PRUs.

Public Data Network (PDN) -
A commercial PSN that supports the interface described in the CCITT protocol X.25.

Queues -
Linked sequences of blocks, tables, messages, and so forth. Most network queues are maintained by leaving the queued elements in place and using tables of pointers to the next queued element. Most queues operate on a first-in, first-out basis. A series of worklist entries for a TIP is an example of a CCP queue.

Random File -
In the context of the NOS operating system, a file with the random bit set in the file environment table; individual records are accessed by their relative PRU numbers.

Record -
(1) A data unit defined for the host record manager; (2) a data unit defined for remote batch workstations. In either case, a record contains space for at least one character of data and normally has a header associated with it. HASP records can be composed of subrecords.

Regulation -
The process of making an NPU or a host progressively less available to accept various classes of input data. The host has one regulation scheme, the host and multiplex interfaces of a local 255x series network processing unit have another scheme, and the multiplex interface to a neighbor NPU has a third regulation scheme. Some types of terminals (for instance, HASP workstations) may also regulate data. Messages are classified as priority data and nonpriority data. Priority of data is established on a device-by-device basis through the PRI classification in NDL.

Remote Batch Facility (RBF) -
The network host product application that supports remote batch job submittal and processing. Remote batch processing can be performed from mode 4, HASP multileaving, 2780, or 3780 protocol terminals.

Remote NPU -
A network processing unit linked indirectly to a host computer through other network processing units. Contrast with Local NPU.

Response Messages -
A subclass of supervisory or service messages that is a response to a supervisory or service message of the originator. Response messages normally contain the requested information or indicate that the requested task has been started or performed. Error or abnormal responses are sent when the responder cannot deliver the information or start the task.

Routing -
The process of sending data/commands through the network to its destination (for instance, a terminal). The network logical address (DN, SN, CN) is the primary criterion for routing. In the NPU, directories are used to accomplish the routing function.

Sequential -
A file organization in which records are stored in the order in which they are generated.

Service Channel -
The network logical connection used for service message transmission. For this channel, the connection number is 0. The channel is always configured, even at load time.

Service Message (SM) -
The network method of transmitting most command and status information to/from the NPU. Service messages use CMD blocks in the internal block protocol.

Service Module (SVM) -
The set of NPU programs responsible for processing service messages. SVM is a part of the BIP.

Short PRU -
A PRU that does not contain as much user data as the PRU can hold and is terminated by a system terminator with a level number. Under NOS, a short PRU defines EOR.

Source -
The terminal or host computer program that creates a message.

Source Node (SN) -
The node that interfaces directly to the source of a network data block.

State Program -
A CCP program written in the state programming language. These programs usually are part of a TIP (some are common to all TIPs) but do not operate on the OPS level. They are reached by a call to the multiplex level or are executed automatically by the multiplex subsystem. State programs process modem signals, input data, and output data. Text processing is primarily performed by state programs.

State Program Tables -
CCP tables used by the multiplex subsystem to locate the next state program to execute.

Station -
A provider and/or recipient of data messages; usually synonymous with a terminal or a grouping of terminal devices. This term is used in various ways within mode 4 communications documentation, as shown in table C-1.

String -
A unit of information transmission. One or more strings compose a record. A string can be composed of different characters or contiguous identical characters.

Subfunction Code (SFC) -
See Function Codes.

Subport -
One of several addresses in a port. In this release of CCP, subport is always equal to 0.

Supervisory Message -
A message block in the host not directly involved with the transmission of data, but which provides information for establishing and maintaining an environment for the communication of data, between the application program and NAM, and through the network to a destination or from a source. Supervisory messages can be transmitted to an NPU in the format of a service message.

Switched Line -
A communication line connected with one network processing unit but able to be connected to any one of several terminals via a switching mechanism, such as a dialed telephone line.

Switching -
The process of routing a message or block to the specified internal program or external destination.

Synchronous -
A transmission in which character synchronization is achieved by recognition of a predefined sync character that precedes the block of data.

System Console Operator -
The operator at the host computer console who satisfies tape requests and performs similar actions. Also known as central site operator.

System Resource Unit (SRU) -
The composite value of central processor time, input/output activity, and memory usage.

Terminal -
An entity, external to the data communication network but connected to it via a communication line, that supplies input to, and/or accepts output from, an application program. In the context of this manual, a terminal is each separately addressable group of devices comprising a physical terminal or station.

Terminal Address -
The hardware address of a mode 4 console, a mode 4C printer, or a 3780 card punch. This term is used in various ways within mode 4 communications documentation, as shown in table C-1.

Terminal Class (TC) -
An NDL parameter and supervisory message field value describing the physical attributes of a group of similar terminals, in terms of an archetype terminal for the group.

Terminal Control Block (TCB) -
A control block within CCP containing configuration and status information for an active terminal. TCBs are dynamically assigned.

Terminal Definition Commands -
A group of commands that allow the operator at the terminal or a host application program to control some of the IVT transforms made by a TIP. These commands can (1) change the terminal characters for breaks and cancel signs, (2) select output page format (such as page width and length, number of padding characters after a line feed or carriage return), (3) designate parity type, terminal class, and other device-related features.

Terminal Interface Program (TIP) -
A portion of the Communications Control Program that provides an interface for terminals connected to a 255x series network processing unit. The TIP performs character conversion to and from 7-bit ASCII, limited editing of the input and output stream, parity checking, and so forth.

Terminal Name (TNAME) -
A name of up to seven letters and digits known to the network and used to identify a device to the network operator.

Terminal Node -
The node number associated with an NPU that interfaces with a terminal.

Terminal Operator -
The person operating the controls of a terminal. Contrast with User.

Terminal Servicing Facility -
See Application Program.

Terminal Verification Facility -
A network application program that provides selectable tests for users to verify proper operation of an interactive terminal.

Test Utility Program (TUP) -
A debugging utility that supports breakpoint debugging of CCP as well as other utility type operations such as loading and dumping.

Timeout -
The process of setting a time for completion of an operation and entering an error processing condition if the operation has not finished in the alloted time.

Timing Services -
The subset of base system programs within CCP which provide timeout processing and clock times for messages, status, etc. Timing services provide the drivers for the real-time clock.

Trailer -
Control information appended to the end of a message unit. A trailer contains the end-of-data control signals. Trailers can be generated by the terminal or by an intermediate device such as a frame generator. Not all headers are matched with trailers, although some devices split their control information between a header and a trailer. The trailer usually contains a data assurance field such as a CRC-16 or a checksum. Like headers, trailers are generated and discarded at various stages along a data unit's path.

Transaction Facility (TAF) -
The network host product application that supports transactional terminal operation. Transactional refers to a terminal operation that is used to conduct a single, simple data base access or retrieval procedure, such as a business transaction. A transactional terminal is distinctive in that the operator is aware of only the transaction being conducted. Terminals used with the Transaction Facility can be special-purpose terminals (such as point-of-sale terminals) or standard asynchronous or mode 4 terminals. When asynchronous or mode 4 terminals are connected to the Transaction Facility, their operation is oriented toward performing data base manipulation through the console.

Transmission Mode -
During input, the network transmission mode is the terminal character code translation and editing option selected by the terminal user; during output, the network transmission mode is the terminal character code translation option selected by the application program. Normalized and transparent mode transmissions are both supported by the network software. Character mode is the initial and default mode for all terminals. Transparent mode should only be used when meaningful to the applicaton program with which communication is occurring.

Transmission Size -
During input, the number of characters and lines reaching the terminal interface program in a single transmission. During output, transmission size should be synonymous with page size (as defined by the terminal user though the page width and page length parameters).

Transparent Mode -
A software feature provided by the Network Access Method and the network processing unit TIP. When transparent mode transmission occurs between an application program and a terminal, the Network Access Method does not convert data to or from display code, and the TIP does not edit the character stream or convert the characters to or from 7-bit ASCII code. When no parity is in effect for the terminal and transparent mode transmission occurs, all eight bits of the character byte can be used to represent characters in 256-character sets (such as EBCDIC).

Trunk -
The dedicated communication line connecting two network processing units.

Trunk Protocol -
The protocol used for communicating between neighboring NPUs. It is a CDC protocol that uses the frame as the basic communication element.

Typeahead -
The ability to transmit many messages in succession to the same application program, without needing to wait for a response to each message. The transmission buffering done by the network software allows many programs to apparently support typeahead of a few messages, when they actually do not support the feature. Typeahead is formally supported by application programs that do not expect the terminal user to wait for a response to one message before entering the next one. When typeahead is supported, the terminal user effectively creates a stack of input messages within the host computer that, when processed in the order entered, perform a series of dependent or independent functions; it is the terminal user's responsibility to recover from unforeseen effects of messages at the beginning of the message stack, based on the results returned to the terminal after the last message in the stack has been processed.

Typeahead (Terminal) -
The ability of a terminal to enter input data at all times. The ASYNC TIP supports typeahead; the X.25 TIP supports typeahead if it is provided by the PSN.

Upline -
The direction of input flow to a host computer application program.

User -
That person or group of people who are the preparers and/or recipients of messages communicated with an application program via the network. A user can interface with one or more terminals, or with no terminals. Contrast with Terminal Operator.

User Name -
The NOS validation file parameter entered by the terminal operator during the Network Validation Facility login procedure.

Virtual Channel (X.25/PAD) -
A channel defined for moving data between a terminal and a host. Virtual channels are defined for the length of time that the terminal is connected to the PSN.

Virtual Circuit -
See Virtual Channel.

Word -
The basic storage and processing unit of a computer. The NPU uses 16-bit words (macromemory) and 32-bit words (internal to the micro processor only). All interfaces are 16-bit words (direct memory access) or in character format (multiplex loop interface). Characters are stored in the 255x series network processing unit macromemory two per word. Hosts (CYBER series) use 60-bit words but a 12-bit byte interface to the NPU.

Some terminals such as a HASP workstation can use any word size but must communicate to the NPU in character format. Therefore, terminal word size is transparent to the NPU.

Worklist Processor -
Within CCP, the base system programs responsible for creating and queuing worklist entries.

Worklists -
Within CCP, packets of information containing the parameters for a task to be performed. CCP programs use worklists to request tasks of OPS level programs. Worklist entries are queued to the called program. Entries are one to six words long, and a given program always has entries of the same size.

X.25 Protocol -
A CCITT protocol used by the packet-switching network. It is characterized by high-speed, framed data transfers over links. A PSN requires a PAD access for attaching asynchronous terminals.

X.25 TIP -
The CCP TIP that interfaces an NPU to a packet-switching network.

Zero-Length PRU -
A PRU that contains system information but no user data. Under NOS, a zero-length PRU defines EOF.

MNEMONICS

Following is a list of mnemonics used in this manual.

ABL	Application Block Limit
ACN	Application Connection Number
ACT	Application Character Type
AIP	Application Interface Program
ANAME	Application Name
APL	A Programming Language
ASCII	American Standard Code for Information Interchange
ASYNC	Asynchronous
BCD	Binary Coded Decimal
BIP	Block Interface Package
BLK	Message Block
BRK	Break Block
BSC	Binary Synchronous Communication
BT	Block Type
B1, B2	User-defined breaks
CA	Cluster Address
CCITT	Comite Consultif International Telephonique et Telegraphique (an international communications standards organization)
CCP	Communications Control Program
CDCCP	Control Data Communications Control Procedure
CDT	Conversational Display Terminal
CE	Customer Engineer
CIB	Circular Input Buffer
CLA	Communications Line Adapter
CMD	Command Block
CND	Connection Number Directory
CR	Carriage Return
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube
CS	Communications Supervisor
DBC	Data Block Clarifier (for blocks/SVM)
DBZ	Downline Block Size
DEL	Delete character
DN	Destination Node
DSR	Data Set Ready
DT	Device Type
EBCDIC	Extended Binary Coded Decimal Interchange Code
EOF	End of File
EOI	End of Information
EOJ	End of Job
EOM	End of Message
EOR	End of Record
EOT	End of Transmission
ETB	End of Transmission Block
ETX	End of Text
FCS	Function Control Sequence (HASP protocol)
FD	Forward Data (block protocol)
FDX	Full Duplex
FE	Format Effector
FET	File Environment Table
FF	Form Feed
FN	Field Number

FS	Forward Supervision (block protocol)
FV	Field Value
HASP	Houston Automatic Spooling Program Protocol
HDLC	High-level Data Link Control
HDX	Half Duplex
HIP	Host Interface Package
HO	Host Ordinal
HOP	Host Operator
IAF	Interactive Facility program
ICMD	Interrupt Command
ICMDR	Interrupt Command Response
ICT	Input Character Type
INITN	Initialization Block Acknowledgment
INITR	Initialization Block Request
ISO	International Standards Organization
IVT	Interactive Virtual Terminal
LCF	Local Configuration File
LF	Line Feed
LFG	Load File Generator
LIP	Link Interface Package
LP	Line printer
MCS	Message Control System
MLIA	Multiplex Loop Interface Adapter
MPLINK	The Pascal link editor
MSG	End-of-message block
MTI	Message Type Indicators (Mode 4 protocol)
NAK	Negative Acknowledgment Block
NAM	Network Access Method
NCB	Network Configuration Block
NCF	Network Configuration File
NDA	Network Dump Analyzer
NDLP	Network Definition Language Processor
NIP	Network Interface Program
NLF	Network Load File
NOP	Network Operator
NPU	Network Processing Unit
NS	Network Supervisor program
NVF	Network Validation Facility
ODD	Output Data Demand (Multiplex subsystem)
PAD	Packet Assembly/Disassembly
PDN	Public Data Network
PFC	Primary Function Code
PIP	Peripheral Interface Program
PL	Page Length (IVT)
PPU	Peripheral Processing Unit
PRU	Physical Record Unit
PRUB	Physical Record Unit Block
PSN	Packet Switching Network
PW	Page Width
QTRM	Queued Terminal Record Manager
RAM	Random Access Memory
RBF	Remote Batch Facility program
RC	Reason Code
RCB	Record Control Byte (HASP protocol)
ROM	Read Only Memory
RR	Receive Ready (trunk or X.25 protocol)
RS	Reverse Supervision (block protocol)
RST	Reset Block
RTS	Request to Send
SAM-P	System Autostart Module Program
SARM	Set Asynchronous Mode (trunk or X.25 protocol)
SCB	String Control Byte (HASP protocol)
SFC	Secondary Function Code
S-Frame	Supervisory Frame (trunk or X.25 protocol)
SRCB	Subrecord Control Byte (HASP protocol)
STX	Start of Text
SVM	Service Module (for processing service messages)
SYNC	Synchronizing Element
TAF	Transaction Facility
TC	Terminal Class
TCB	Terminal Control Block
TIP	Terminal Interface Program
TNAME	Terminal Name
TO	Timeout

TTY	Teletypewriter
TUP	Test Utility Program
TVF	Terminal Verification Facility
UA	Unnumbered Acknowledgment (trunk or X.25 protocol)
UBZ	Upline Block Size
U-Frame	Unnumbered Frame (see UA and UI)
UI	Unnumbered Information frame (trunk or X.25 protocol)
US	Unit Separator
XBZ	Transmission Block Size
X-OFF	Stop character
X-ON	Start character
XPT	Transparent
X.3	CCITT protocol for asynchronous terminal access to a packet-switching network
X.25	CCITT protocol for packet-switching networks
X.28	CCITT protocol for terminal access to PSN/PAD
X.29	CCITT protocol for host access to PSN/PAD

Figure D-1 duplicates a thermal printer listing of an actual terminal session. The terminal user logged in using the abbreviated full login procedure, then entered TIP commands to control the network functions during input and output. The application program used is site-written.

TUTOR sends and receives data in ASCII code and does not support transparent mode input or output. Attempting to use transparent mode while communicating with TUTOR causes TUTOR to issue its diagnostic display. Use of TUTOR from a mode 4 BCD terminal is possible, but the resulting output is completely capitalized, lines are folded awkwardly, and some of the punctuation is altered by the character code translation involved during output. TUTOR cannot be used from a HASP terminal.

```
 cr <------------------------------------------------ Type-in for line speed auto recognition.
 cr <------------------------------------------------ Type-in for character set auto recognition.

NO HOST SELECTED      CONTROL CHARACTER = (ESC)
NPU NODE = 01         TERMINAL NAME = TM1016
HOST        NODE      SELECTED/    STATUS
                      CONNECTED
SVLNOS1     12                     AVAILABLE           Terminal is not configured for automatic
SVLNOS1     22                     NOT AVAILABLE  <--- host selection, but is configured with HD=Y
SVLNOS2     32                     AVAILABLE           initially, so that the complete host availability
                                                       display appears.
ENTER <CT> HN=NN TO SELECT HOST
   OR <CT> HS=HOSTNAME

 HS=SVLNOS1 cr <------------------------------------- Terminal definition command to select a host
                                                       by name.
HN ACCEPTED ..  <----------------------------------- Response from network software.

 cr <------------------------------------------------ Empty line to cause connection to selected
                                                       host; needed because this terminal is not
                                                       configured for intial automatic connection
                                                       to a host.

WELCOME TO THE NOS SOFTWARE SYSTEM. <---------------- Physical connection to the selected host is
COPYRIGHT CONTROL DATA 1978, 1985.                     completed.  Login can begin.

85/03/04. 07.26.59. TM1016
(22) SVL SN614 NOS                          NOS2-8J01T/R3B.  <----- NVF banner prompt for login.
FAMILY:  cr <--------------------------------------- Long form login dialog begins.
 ct=% cr <------------------------------------------ Terminal definition command to preserve terminal
                                                       definition command flag character.

CT ACCEPTED ..  <----------------------------------- Response from network software.

%ci=9 cr
      Terminal definition commands to preserve beginning of print lines during output.
CI ACCEPTED ..  <----------------------------------- Response from network software.

%li=2 cr

LI ACCEPTED ..  <----------------------------------- Response from network software.

USER NAME:  usr4     cr
PASSWORD:   pass  cr
TM1016  -  APPLICATION:  tutor  cr
```

Figure D-1. Sample Terminal Session (Sheet 1 of 6)

```
You are communicating with the network application program called
TUTOR.  If output of a TUTOR display stops without requesting input
from you, or if OVER.. appears, enter an empty line to continue.

TUTOR associates the following characteristics with your terminal:

  Terminal class of 2
  Terminal name of TM1016
  Page width of 72 characters per line
  Page length of 0 lines
  Block size of 130 characters
  Block limit of 2 blocks
  Famile name in use is SYS5 with the ordinal 0003
  User name in use is USR4 with the index 0067

These are the only terminal characteristics directly known to any
network application program.  The network software associates other
characteristics with your terminal, according to the way the
terminal name was configured.

If you want more information, type yes; if you want to end communica-
tion with TUTOR, type no.  Please enter your answer without leading
blanks or any other characters in the same line:

%p1=15 [cr] <-------------------------------------  Terminal definition command to format page cor-
                                                    rectly during page wait output.

PL ACCEPTED .. <----------------------------------  Response from network software.

yes [cr] <----------------------------------------  Response to TUTOR prompting line.

Thank you.  TUTOR can provide information on the following topics:

      (1) Network application programs
      (2) Changing terminal characteristics
      (3) Login procedures and switching network application programs

If you want information on one of these topics, type the number given
above for the topic but do not enter the parentheses; if you want to
end communication with TUTOR, type no.  Please enter your answer with-
out leading blanks or any other characters in the same line:

1 [cr] <------------------------------------------  Response to TUTOR prompting line.

%pg=y [cr] <--------------------------------------  Terminal definition command turns page wait on
                                                    because display selected will be long.

PG ACCEPTED ..  <---------------------------------  Response from network software.

Network application programs are a special group of programs that
provide services to terminal users.  Only programs identified in the
following list are directly accessable by a terminal user:

      Remote Batch Facility (called RBF)
      Interactive Facility (called IAF)
      Transaction Facility (called TAF)
      Terminal Verification Facility (called TVF)
      This program (called TUTOR)
      Network Validation Facility (called NVF)

Any other host computer software you need to use can be accessed
through either the Remote Batch Facility or the Interactive Facility.

%b1=: [cr] <--------------------------------------  Terminal definition command makes user-break-1
                                                    character visible and acts as a page turning
                                                    response.

B1 ACCEPTED ..  <---------------------------------  Response from network software.
```

Figure D-1. Sample Terminal Session (Sheet 2 of 6)

```
Network Validation Facility

NVF performs login validation processing.  It also gives you a way to
switch terminal communications from one network application program
to another.  NVF is the only network application program you do not
need to request communication with; your terminal is automatically
given as much access to NVF as necessary.  In response to the
prompting message APPLICATION:  from NVF, you enter the name of the
network application program you want to use.  The name you enter must
have the form shown in parentheses at the beginning of this display;
however, you cannot request NVF to connect you with itself (you
cannot enter NVF as a response to an APPLICATION:  prompt.)

[cr] <-------------------------------------------- An empty line is entered to acknowedge the page.
                                                    TUTOR does not see this entry.

You can encounter many variations of a basic dialog with NVF.  The
form of login you use is partially determined by the way your terminal
name is configured.  The general form of this login dialog is described
in each of the reference manuals for network application programs.

Interactive Facility

IAF comprises six subsystems that permit you to create files and
create or execute programs from your terminal, without requiring the
use of card readers or line printers.  These subsystems are:

     BASIC; this subsystem is an interactive version of the interpreter

%b2=) [cr] <---------------------------------------- Terminal definition command to make user-break-2
                                                    character visible also turns the page.

B2 ACCEPTED ..  <----------------------------------- Response from network software.

     for the BASIC programming language.

     FTNTS; this subsystem is an interactive version of the FORTRAN
     Extended 4 compiler.

     FORTRAN; this subsystem is an interactive version of the
     FORTRAN 5 compiler.

     EXECUTE; this subsystem provides a quick method to run frequently
     used programs that you have previously stored in the system in
     object-code form.

     NULL; this subsystem provides a quick method to manipulate files
     and perform certain other operations.

[cr] <-------------------------------------------- Page acknowledgment only.

     BATCH; this subsystem permits you to enter and execute operating
     system control statements one at a time.  You can use any software
     product for which a control statement exists through this
     subsystem, or execute any application program through this
     subsystem.

Another subsystem, the ACCESS subsystem, is available to some users.
This subsystem allows communication from one terminal connected to
IAF, to another terminal connected to IAF.

To switch your terminal from one subsystem to another within IAF, you
type the name of the subsystem you want to use.  To switch your terminal
from a subsystem to another network application program, you must

[cr] <-------------------------------------------- Page turn.
```

Figure D-1. Sample Terminal Session (Sheet 3 of 6)

```
disconnect from IAF by typing BYE, comma, and the name of the
network application program you want to communicate with.

Remote Batch Facility

RBF permits you to enter a job file from a remotely located card reader
and receive job output at remotely located batch devices.  The job's
progress through the host system can be tracked from your terminal and
the job's files can be manipulated before and during output.  Your
terminal can be switched directly from RBF to IAF by typing IAF, or
from RBF to NVF (to request connection with another network application
program) by typing END.

[cr] <---------------------------------------------  Page turn.  Note that the page length calculation
                                                     includes blank lines.

Transaction Facility

TAF permits you to access a database specifically created to treat any
operation performed by your terminal as a transaction that reads or
alters that database.  Very little dialog is involved when using TAF.

Terminal Verification Facility

TVF provides several tests you can select to verify that your terminal
is sending and receiving data correctly.  Your terminal can be switched
from TVF to NVF by typing END.

[cr] <---------------------------------------------  Page turn.

This concludes TUTOR's display of application programs.  If you want to
select another display, type yes and you will be returned to the
display of available options; if you want to end communication with
TUTOR, type no.  Your choice is:

4 [cr] <-------------------------------------------  User responds to prompt with an option not listed
                                                     for this display.  TUTOR treats as an error.

The line you have entered is not recognized by TUTOR.  Please enter
your choice again, following these rules:

      Your entry is the first and only thing you enter on the line.

      Your entry is correctly spelled and is given in a TUTOR display
      as a word to be typed in.

      Your entry is all uppercase or all lowercase characters.

The display appearing when you made your entry will not be repeated.
The following entries are valid:

  yes     selects the display of available options

[cr] <---------------------------------------------  Page turn.

  no      selects the display of communication-ending options

  1       selects the display of network application programs
  2       selects the display for changing terminal characteristics
  3       selects the display of login procedures

  hello   selects the beginning of a new login procedure,
  login   after performing an automatic logoff for your terminal

  bye     selects network software disconnection processing for
  logout  your terminal and causes it to be logged off

  iaf     selects communication with the Interactive Facility
  rbf     selects communication with the Remote Batch Faility

[cr] <---------------------------------------------  Page turn.
```

Figure D-1. Sample Terminal Session (Sheet 4 of 6)

```
  taf     selects communication with the Transaction Facility
  tvf     selects communication with the Terminal Verification Facility

Please enter your choice on the next line:

  %pg=n [cr] <-------------------------------------  User knows what the mistake was.  To speed output
                                                     of the next display, page wait is turned off.

PG ACCEPTED ..  <--------------------------------  Response from network software.

Next display is omitted, except for following line:

and receive job output a■) [cr] <----------------  User enters break character, network software
                                                     discards remainder of current display, and TUTOR
                                                     interprets as a request to repeat the menu display.

Repeated display is omitted.

no [cr] <-----------------------------------------  User decides to end TUTOR session.

Thank you.  You have chosen to end communication with TUTOR.  Your term-
inal will now be switched to communicate with the Network Validation
Facility program.  If you want NVF to prompt you for the name of the
next network application program, type end without leading blanks or
any other characters in the same line; if you do not want to be prompt-
ed, you can enter any of the following instead:

      IAF     selects the Interactive Facility program
      RBF     selects the Remote Batch Facility program
      TAF     selects the Transaction Facility program
      TVF     selects the Terminal Verification Facility program

      BYE     selects network software disconnection processing for
      LOGOUT  your terminal and causes it to be logged off

      HELLO   selects the beginning of a new login procedure, after
      LOGIN   performing an automatic logoff for your terminal

Your choice is:

iaf [cr] <----------------------------------------  TUTOR supports direct switches between network
                                                     application programs.

TUTOR CONNECT TIME 00.10.33.
TERMINAL:  26, NAMIAF
RECOVER/ CHARGE:  charge,1234,567890 [cr] <-------  User responds to IAF prompting.  Notice that input
                                                     line begins on same line as output ended.

$CHARGE,1234,567890.

/bye,logoff [cr] <--------------------------------  User enters this application-switching command
                                                     from memory.  IAF accepts BYE, and passes LOGOFF
                                                     as an application name to NVF.

USR4      LOG OFF       07.41.56.
USR4      SRU         1.002. UNTS.

IAF       CONNECT TIME 00.01.57.
ILLEGAL APPLICATION, TRY AGAIN.  <----------------  NVF does not recognize LOGOFF.
TM1016 - APPLICATION:  logout [cr] <--------------  User reenters corrected command to NVF directly.
LOGGED OUT.
```

Figure D-1. Sample Terminal Session (Sheet 5 of 6)

```
NO HOST CONNECTED      CONTROL CHARACTER = %
NPU NODE = 01          TERMINAL NAME = 8J01T
HOST          NODE     SELECTED/  STATUS
                       CONNECTED
SVLNOS1       12       SA         AVAILABLE
SVLNOS1       22                  NOT AVAILABLE
SVLNOS2       32                  AVAILABLE
ENTER INPUT TO CONNECT TO HOST

%CH [cr] <----------------------------------------
TC=2, BS=(BS), CN=(CAN), AB=(CAN), B1=(, B2=), <----
EL=(CR), EB=(EOT)
.  <----------------------------------------------
```

Disconnection from the host is indicated by the host availability display.

Terminal definition command to display effects of changes made during the session.

Response from network software.

Disconnection physically occurs after 2 minutes of inactivity.

Figure D-1. Sample Terminal Session (Sheet 6 of 6)

INDEX

AB command 5-19
Abbreviated login 4-19
Abort-output-block
    Command 5-19
    Use 2-20
Access procedures 4-1
Address C-1
Administrative interruptions 4-22
APL
    Character sets 5-32, A-1, C-1
    Code sets A-1
Application
    Block header C-1
    Block limit C-1
    Block number C-1
    Block type C-1
    Interface program C-1
    Name 4-18, C-1
    Program 1-7, 4-16, C-1
AR command 5-33
Archetype terminal C-1
ASYNC TIP 6-7
Asynchronous terminals
    Communication lines 3-1, C-1
    Feature dependencies 3-4
    Terminals 1-3, 3-1, C-1
Automatic
    Application selection 4-16
    Input C-1
    Login procedure 4-12, C-1
    Recognition command 5-33
    Recognition procedure 4-7, C-2

Backspace
    Character 2-18, 5-7
    Command 5-18
    Use 2-17
Base system software C-2
Batch
    Data 2-6, 6-17, 6-23, 6-29, 6-35
    Device C-2
    Mode C-2
    Terminals 1-4, 3-16, 3-18, 3-21
BF command 5-39
Bisynchronous
    Communication lines 3-1, C-2
    Terminals 1-4, 3-20, 6-27
Block
    Acknowledgment C-1
    Definition C-2
    Limit C-2
    Mode 2-3
    Network 2-1, 6-4
    Sizes 2-2, 2-13
    Terminal transmission 2-1
BR command 5-24
Break
    Characters 2-19, 5-23, 5-24
    Definition C-2
    Key 2-20
    Significance 6-11, 6-15
BS command 5-17
BSC TIP 6-27
Buffering C-2
BYE 4-21

Bypassing input queue 5-22
Byte C-2
B1 command 5-23
B2 command 5-24

Cancel
    Character 2-19, 5-18
    Command 5-18
    Input line 2-19
Carriage control operations
    HASP printers 3-21
    Mode 4 printers 3-18
    2780/3780 printers 3-24
    3270 batch printers 3-26
Carriage return
    Idle count 5-25
    Keys xiii, 2-4
    Operation 2-2
Cassette drive 6-2, C-2
CCP (See Communications Control Program)
CE Error message C-2
CH command 5-14
Change
    Abort-output-block character 5-19
    Backspace character 5-17
    Blocking factor 5-39
    Cancel-input-line character 5-18
    Character set 5-33
    Control character 5-15
    Echoplexing 5-48
    Editing 5-34, 5-35
    Host 4-11
    Idle character count 5-25, 5-26
    Input control 5-21
    Input device 5-49
    Output control 5-20
    Output device 5-50
    Page length 5-30
    Page wait 5-27
    Page width 5-29
    Parity 5-46
    Punch card translation 3-17
    Terminal class 5-15
    Transparent mode delimiters 5-42, 5-44
    User-break-1 character 5-23
    User-break-2 character 5-24
Character
    Code conversions 2-11
    Code sets A-1
    Definition C-2
    Mode C-3
    Secure login 4-14
    Sets 2-9, 2-10
CI command 5-25
Circular input buffer C-3
Cluster C-3
Cluster address C-3
CN command 5-18
Command driver C-3
Command summary inside front cover
Communication
    Breaks 4-22
    Failures 4-22
    Lines 1-3, 4-1, C-3
    Path 4-1

Communications Control Program (CCP) 6-1, C-3
Communications Supervisor (CS) C-3
CONFIG card 4-9, 6-26
Connection number C-3
Connection regulation 6-6
Console
    Data 2-6
    Description 1-4
    Exchange modes 2-8
    Input concepts 2-2
    NPU 6-2
    Output concepts 2-6
Constant carrier C-3
Contention C-3
Control blocks C-3
CONTROL key xiii
Controlled carrier C-3
Controlled terminal C-3
Conversions
    Host software input 2-24
    Host software output 2-24
    NPU input 2-24
    NPU output 2-12
    Terminal input 2-23
    Terminal output 2-25
Coupler C-3
CP command 5-31
cr xiii
CROSS C-4
CS (See Communications Supervisor)
CT command 5-15
Cursor positioning 3-17, 5-31
CYBER channel coupler 6-1, C-3
Cyclic redundancy check (CRC) C-4

Data
    Buffer C-4
    Compression 6-28, C-4
    Definition C-4
    Priorities 6-6
    Set 4-5, C-4
Dedicated line C-4
DEFINE C-4
Destination C-4
Destination Node (DN) C-4
Device
    Control 3-7
    Definition C-4
    Options 3-8
Diagnostic messages B-1
Dial-in procedure 4-5
Dialog
    Terminal Interface Program B-1
    Terminal Verification Facility 7-1
Direct transmission terminal 2-14
Discarding output 2-19
Disconnection
    Application program 4-21
    Network 4-21
    Procedures 4-21
    Terminal Verification Facility 7-4
Display code C-4
Display console 1-5
DL command 5-42
DN (See Destination Node)
Downline C-5
Dump C-4
Duplex (see Echoplexing)

EB command 5-41
Echo C-4
Echoplexing
    Control 5-48
    Definition C-4
    Locations 1-4
Editing
    Complete 2-9
    Full ASCII 2-10, 5-35
    Keys 2-15
    Special 2-9, 5-34
    Terminal 2-4
EL command 5-40
END 4-20, 7-4
ENDL 7-3
End-of-block indicator 2-6, 5-41
End-of-line indicator 2-6, 5-40
EP command 5-48
Error recovery 6-11, 6-15, 6-20, 6-27, 6-33, 6-36
ESCAPE 5-2
Exception-processor terminal 2-14
Exit application program 4-20
EX.LOGT 4-20

FA command 5-35
Facility
    Interactive Facility (IAF) 1-7, 4-18, 4-19, 4-20
    Network Validation Facility (NVF) 1-7, 4-18
    NOS/Interactive Facility 1-7, 4-18, 4-19, 4-20, 4-21
    Remote Batch Facility (RBF) 1-7, 4-18, 4-19, 4-20, 4-21
    Site-written 1-7
    Terminal Verification Facility (TVF) 1-9, 4-18, 4-19, 4-20, 4-21, 7-1
    Transaction Facility (TVF) 1-9, 4-18, 4-19, 4-20, 4-21
File
    Definition C-4
    Registers in NPU C-4
Format
    Changing 5-24
    Effector
        Asynchronous terminals 3-9
        Definition C-5
        HASP printers 3-20
        Mode 4 printers 3-18
        X.25 consoles 3-9
        2780/3780 terminals 3-24
        3270 terminals 3-26
    Input 2-2
    Output
        Asynchronous terminals 3-8
        HASP terminals 3-19
        Mode 4 terminals 3-17
        X.25 terminals 3-15
        2780/3780 terminals 3-23
        3270 terminals 3-23
    Terminal definition command 5-12
Frame 6-13, C-5
Free-wheeling terminal C-5
Front-end NPU C-5
Full ASCII editing 2-10, 5-35
Full duplex C-5
Function codes C-5

GOODBYE 4-21
Graphic characters 3-1, 3-14

HAD (see Host Availability Display)
Half duplex C-5
Halt codes C-5
HASP
    Format effectors 3-20
    Protocol 1-4, 3-1, C-5
    Stream numbers 3-19
    Terminals 3-17
    TIP 6-21
    Workstation setup 3-18
HC command 4-12
HD command 4-12
Header C-5
HELLO 4-20
HELP 7-1
High-speed synchronous line C-5
HN command 4-11
Host
    Availability Display (HAD) 4-12
    Definition C-5
    Interface Program (HIP) C-5
    Node 4-11, C-5
    Operator (HOP) 4-22, C-5
HS command 4-10

IAF (see Interactive Facility)
IBM host features 3-19, 3-21
IC command 5-21
IN command 5-49
Information
    Flow 1-6, 2-1
    Structures 2-1
Initialization C-5
Input
    Buffer C-6
    Conversion 2-22
    Definition C-6
    Demultiplexing 6-3
    Editing keys 2-16
    Multiplexing 6-3
    Processing 6-5
    Regulation 6-11, 6-15, 6-20, 6-26, 6-33
    Transmissions 2-13
Interactive device C-6
Interactive Facility (IAF) 1-7, C-6
Interactive mode C-6
Interactive terminals 1-4, C-6
Interactive Virtual Terminal (IVT) C-6
Interrupt key 2-19, C-6
Interruptions in communication 4-22

LCF (see Local Configuration File)
Level C-6
LF xiii
LI command 5-26
Line
    Canceling 2-19
    Communication 1-3, 4-2
    Definition C-6
    Feed 2-4, 3-7, 5-26
    Input 2-2
    Mode 2-2
    Output 2-7
    Status 6-7
    Test 7-3
    Turnaround 2-7
Line feed
    Idle character count 5-26
    Keys 2-4, 2-15
Link 6-12, C-6
Link Interface Program (LIP) C-6
LK command 4-22
Load C-6
Local batch users A-3
Local configuration file (LCF) C-6
Local NPU C-6
Locked keyboards 2-19
Logical
    Connection C-6
    Data structures 2-1
    Lines 2-3, 2-8, C-6
    Link C-6
    Link regulation 6-6
    Record C-6
    Terminal C-7
Login
    Abbreviated 4-19
    Automatic 4-14
    Commands 4-19
    Dialog 4-16
    Problems B-1
    Secure 4-14
LOGOFF (see Logout)
Logout
    With disconnection 4-21
    Without disconnection 4-20
Loop multiplexer C-7
Loopback test 7-2
Low/Medium-speed voice-grade line C-7

Macromemory 6-1, C-7
MCS (see Message Control System)
Message
    Canceling 2-19
    Definition C-7
    Input 2-1
    Output 2-7
Message Control System (MCS) 1-9, 4-18, 4-19, 4-20, 4-21
Micromemory 6-1, C-7
Microprocessor C-7
MLIA (See Multiplex loop interface adapter)
Mode
    Batch 2-1
    Block 2-1, 3-1
    Conversational 2-1
    Normalized 2-11
    Transparent 2-11
Mode 4
    A terminal 1-4, 3-16
    Autoloading 3-16
    C terminal 1-4, 3-16
    Carriage control operations 3-18
    Definition C-7
    Format effector operations 3-18
    Switch setting 3-16
    TIP 6-16, C-7
Modem 4-5, C-7
Monitor C-7
MS command 4-22
Multiple logical lines 2-6
Multiple message transparent mode 2-12, 5-44
Multiplex loop interface adapter (MLIA) 6-2, C-7
Multiplex subsystem 6-2, C-7

NCF (see Network Configuration File)
NDL (see Network Definition Language)
Neighbor NPUs C-7
Network
    Access Method (NAM) C-7
    Address C-7
    Application programs 1-7
    Blocks 2-1
    Configuration C-7

Network (Contd)
Configuration file (NCF) C-7
Definition C-7
Definition file C-8
Definition Language (NDL) C-8
Host software 1-6
Network operator
Communication with 4-22
Functions 1-2
Network processing unit (NPU)
Definition C-8
General functions 1-6
Specific functions 2-2, 6-1
Network structure 1-1
Network Supervisor (NS) C-8
Network Validation Facility (NVF) 1-7, 4-18, C-8
Node C-8
Normalized mode
Definition 2-8, C-8
Input conversions 2-24
Operation 2-8
Output conversions 2-24
NOS/VE Interactive Facility (VEIAF) 1-9, 4-18, 4-19, 4-20, 4-21
NPU operator C-8
NS (see Network Supervisor)
NVF (see Network Validation Facility)

OC command 5-20
Off-line diagnostics C-8
On-line diagnostics C-8
OP command 5-50
OPS monitor C-8
Output
Buffer C-8
Definition C-8
Demultiplexing 6-4
Device selection 5-50
Interrupting 2-19
Multiplexing 6-3
Processing 6-5
Transmissions 2-19
Owning console C-8

PA command 5-45
Packet C-8
Packet Assembly/Disassembly (PAD) Service C-8
Packet-Switching Network (PSN) C-8
PAD subTIP 6-12, C-8
Page length 2-11, 5-30
Page wait 5-27
Page width 2-11, 5-29
Paging 2-21, 5-27, C-9
Paper tape
Examples 3-4
Input preparation 3-2
Input selection 5-49
Operations 3-1
Output selection 5-50
Parity 2-12, C-9
PASCAL C-9
Passive device C-9
Password 4-16, C-9
PDN (see Public Data Network)
Peripheral Processor Unit (PPU) C-9
PFC (see Primary Function Code)
PG command 5-27
Physical
Data structures 2-1
Lines 2-3, 2-8, C-9
Link C-9
Physical (Contd)
Record unit (PRU) C-9, C-10
Terminal C-9
PL command 5-30
PLATO NAM Interface (PNI) 1-9, 4-18, 4-19, 4-20, 4-21
Polling C-9
Port C-9
Primary Function Code (PFC) C-9
Printer Support Utility (PSU) 1-7, 1-8, 1-9
Priority C-9
Procedure for IBM 3270 terminals 4-10
Protocols 1-3, 3-1, C-9
PRU block (PRUB) C-10
PRU commands C-10
PRU device C-10
Public Data Network (PDN) C-10
Punch selection code 3-20
Punched cards 3-17
PW command 5-28

Queues C-10

Random files C-10
RBF (See Remote Batch Facility)
RC command 4-21
Record C-10
Regulation
Connection 6-6
Definition C-10
Logical link 6-6
Remote Batch Facility (RBF) 1-7, 4-18, 4-19, 4-20, 4-21, C-10
Remote NPU C-10
Reset 6-15
Response message C-10
Routing C-10

Screen test 7-3
SE command 5-34
Secure login 4-14
Sequential C-10
Service channel C-10
Service message (SM) C-10
Service module (SVM) C-10
Short PRU C-10
Signoff 3-19
Signon 3-19, 3-22
Single message transparent mode input 2-12
Site-written application program 1-3
Slave devices 3-7, 3-15
SM (See Service Message)
Source C-10
Source node (SN) C-10
Special editing 2-19, 5-34
SRU (See System resource unit)
State program C-10
State program tables C-11
Station C-11
Stream numbers 3-18
String C-11
Subfunction code (SFC) C-11
Subport C-11
Supervisory message C-11
SVM (See Service module)
Switch settings 3-16, 4-4
Switched line C-11
Switching programs 4-19
Synchronous C-11
System console operator C-11
System resource unit (SRU) C-11

TAF (see Transaction Facility)
TC command 5-15
TCB (See Terminal control block)
Terminal
    Address C-11
    Archetype C-1
    Asynchronous 1-3, 3-1
    Batch 1-4
    Bisynchronous 1-4, 3-20
    Classes 3-1, 5-1, 5-2, 5-15, C-11
    Communication software 1-3
    Control block (TCB) C-11
    Definition commands 5-1, C-11
    Direct transmission 2-14
    Exception processor 2-14
    Feature dependencies
        Asynchronous terminals 3-4
        HASP terminals 3-19
        Mode 4 terminals 3-16
        X.25 terminals 3-14
        2780/3780 terminals 3-21
        3270 terminals 3-23
    HASP 1-4, 3-18
    Input control keys 2-4
    Installation options
        Asynchronous terminals 3-8
        X.25 terminals 3-15
    Interactive 1-4
    Mode 4 1-4, 3-16
    Name (TNAME) C-11
    Node C-11
    Operation 3-1
    Operator C-11
    Output function keys 2-20
    Sample session D-1
    Servicing facility C-11
    Setup 4-4
    Transmission blocks 2-1
    Transparent mode 3-22
    X.25 1-3, 3-14
    3270 3-23
Terminal Interface Program (TIP)
    Commands 5-1
    Definition C-11
    Functions 2-1
    Operation 6-7
Terminal Verification Facility (TVF) 1-9, 4-18, 4-19, 4-20, 4-21, 7-1, C-11
TEST 7-2
Test Utility Program (TUP) C-11
Timeout
    Connection 4-23, C-11
    Login 4-16, B-1
Timing services C-12
TIP (See Terminal Interface Program)
TM command 4-21
TNAME (See Terminal name)
Trailer C-12
Transaction Facility (TAF) 1-9, 4-18, 4-19, 4-20, 4-21, C-12
Transmission
    Blocks 2-1, 3-23
    Key 2-3
    Mode C-12
    Protocols 1-3, 3-1
    Size 2-1, C-12
Transparent
    Definition C-12
    Delimiters 5-2, 5-42, 5-44
Transparent (Contd)
    Input conversion 2-24
    Mode operations 2-11, A-14
Trunk
    Definition C-12
    Multiplexing 6-4
TVF (see Terminal Verification Facility)
Typehead C-12

Upline C-12
User C-12
User-break-1
    Character 5-23
    Function 2-21, 5-23
User-break-2
    Character 5-24
    Function 2-21, 5-24
User name 4-16, C-12

VEIAF (see NOS/VE Interactive Facility)
Virtual channel C-12
Virtual circuit C-12

Word C-12
Worklist C-13
Worklist processor C-13

XL command 5-44
X.25
    Packet assembly/disassembly (PAD) service C-13
    PAD subTIP 6-12, C-13
    Terminals 3-14
    X25 subTIP 6-12

Zero-length PRU C-13

2551 series communications processor 6-1
2741 terminals 3-8
2780/3780
    Automatic recognition procedure 4-10
    Definition 3-20
    TIP 6-27
3270
    Format effector operations 3-26
    Terminals 3-23
    TIP 6-33
    Procedures 4-10
    Transparent mode operation 2-11

/*CONFIG 4-9, 6-26
/*EOI 6-24, 6-29
/*EOR 6-23, 6-29
/*SIGNOFF 6-26
/*SIGNON 6-26

xiii
<--- xiii

CONTROL DATA CORPORATION
Publications and Graphics Division
P. O. BOX 3492
SUNNYVALE, CALIFORNIA 94088-3492

DATE July 31, 1986

TITLE Network Access Method Version 1/Communications Control Program Version 3
Terminal Interfaces Reference Manual

PUBLICATION NO.. 60480600

REVISION. F

REASON FOR CHANGE

Revised to reflect release of NAM Version 1.8 under NOS 2.5.1 at PSR level 664. Miscellaneous technical changes are included.

INSTRUCTIONS

To update this manual, for which the previous revision was E, make the following changes.

Remove	Insert
Front cover/ inside front cover	Front cover/ inside front cover
Title page/ii	Title page/ii
iii/iv	iii/iv
v/vi	v/vi
vii/viii	vii/viii
ix thru xii	ix thru xii
2-11/2-12	2-11/2-12
4-1 thru 4-8	4-1 thru 4-8
4-11 thru 4-20	4-11 thru 4-20
7-1/7-2	7-1/7-2
Index-1 thru -5	Index-1 thru -5
Comment sheet/mailer	Comment sheet/mailer