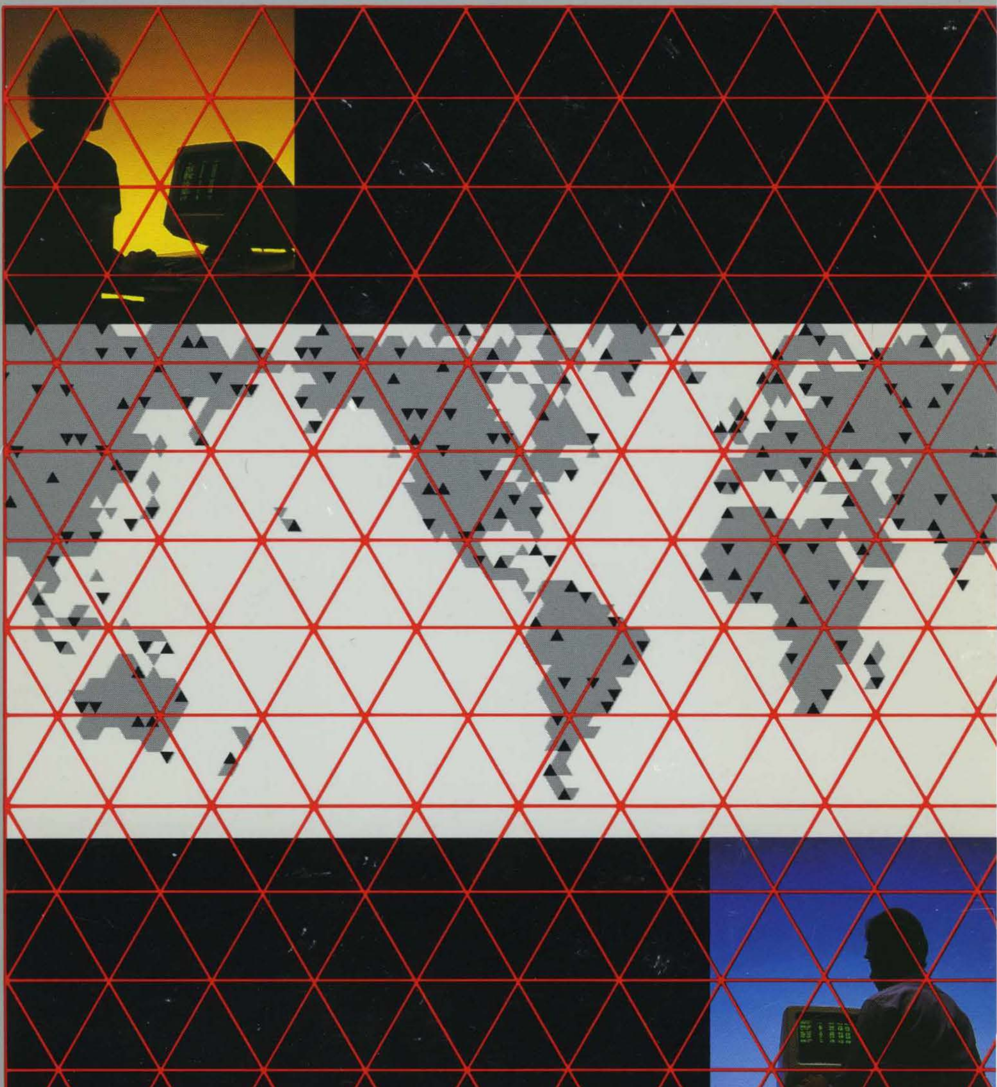
CDCNET Terminal Interface

Usage



60463850

CONTROL DATA

CDCNET

Terminal Interface

Usage

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60463850

Manual History

Revision	System Version PSR Level	Date
A	1.0/647	December 1985
B	1.2/678	April 1987
C	1.2.5/688	September 1987
D	1.3/700	April 1988
E	1.4/716	December 1988
F	1.5.1/739	December 1989

This manual is revision F, printed in December 1989, reflecting CDCNET version 1.5.1 at PSR level 739, for operation on NOS version 2.7.1 and NOS/VE version 1.5.1. It includes the following changes and additions:

- New connection attributes: ATTENTION_CHARACTER_ENABLE, ECHO_ENABLE, INPUT_FLOW_CONTROL_ENABLE, OUTPUT_FLOW_CONTROL_ENABLE, PARITY_ENABLE, and TRANSPARENT_PROTOCOL_MODE have been added to chapter 4.
- A new gateway, the X.25 interactive terminal gateway, is documented in chapter 8.
- Resolving Communications Problems, previously chapter 8, is now chapter 9.
- Added new SCL command, DISPLAY_TERMINAL_MODEL to appendix D.
- Updated tables in appendix E.
- Technical corrections have been made throughout the manual. Change bars indicate altered text.

Contents

Figures

Tables

About This Manual

This manual describes how to access Control Data services through the CDC® Control Data Distributed Communications Network (CDCNET).

Audience

If you are an application programmer, a microcomputer user, a site administrator, or a customer-service analyst who needs to know details about CDCNET's terminal user interface, this manual will be useful to you. The degree of detail in this manual is intended to give you maximum control over CDCNET's terminal user interface. As such, it is not intended for the average terminal user, but for someone providing support for the terminal user's activities.

Prerequisites

This manual assumes that:

- You have detailed information about the terminals you intend to connect to the network.
- You are experienced in your technical areas, but might not be familiar with CDCNET.
- You might come from diverse operating system environments.

NOTE

If you are a TELNET user, consult the CDCNET TCP/IP manual.

Terminology

Terminal in this manual means any interactive device used to access a service through CDCNET. Information on batch devices is described in the CDCNET Batch Device User Guide.

The term **network** is used interchangeably with CDCNET.

Organization

This manual contains introductory, quick-reference, and explanatory materials that are organized as follows:

- Chapter 1 introduces you to basic network concepts as they apply to CDCNET.
- Chapter 2 contains quick-reference descriptions of CDCNET terminal user commands. These include commands to establish new service connections, switch between existing connection, change and display terminal and connection attributes, and obtain help in using CDCNET.
- Chapters 3 and 4 describe CDCNET's terminal and connection attributes, respectively. Terminal attributes are provided so that you can describe the physical aspects of your terminal to the network. Connection attributes are provided so that your application can select a specific style of operation for your session.
- Chapter 5 describes virtual line mode (VLM), transparent mode, and flow control. CDCNET treats your input and output differently, depending upon which mode your connection is using, and whether or not flow control is being used.
- Chapter 6 describes user interrupts processed by CDCNET.
- Chapter 7 describes the interactive terminal passthrough feature.
- Chapter 8 describes the X.25 interactive terminal gateway feature.
- Chapter 9 contains notes and cautions that can help you to resolve terminal-related communications problems.

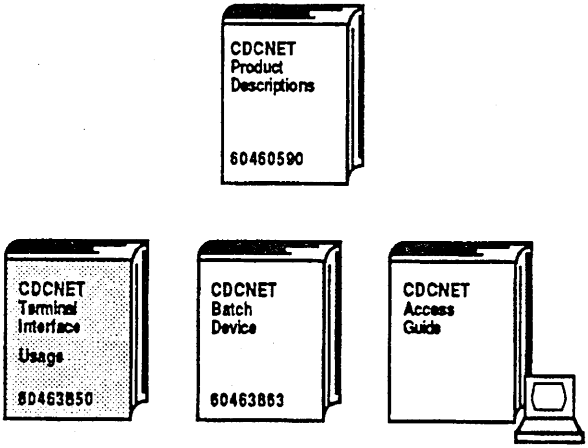
In addition to the chapters listed above, this manual contains ten appendixes. Appendixes A, B, and C provide general reference information: a glossary, the ASCII coded character set, and terminal interface messages. Appendixes D and E supply service-specific information for NOS/VE and NOS users. Appendixes F, G, H, I, and J discuss unique features of the following protocols: 3270 Binary Synchronous Communications Protocol, X.25 Asynchronous Protocol, X.PC Protocol, Mode 4 Protocol, and 3270 SNA Communications Protocol.

Conventions

The following conventions are used in this manual.

boldface	Command names and required parameters are shown in boldface type when illustrating a format. Chapter 2 describes the syntax of CDCNET commands.
italics	Optional parameters (in some cases, called attributes) in quick-reference command descriptions are shown in italics.
boolean values	ON and OFF represent boolean values in quick-reference descriptions and examples.
`examples`	Examples of user entries and computer responses are shown in a font that resembles computer output. These examples use abbreviated commands. Unless specified for a particular terminal, you can recreate the examples on any terminal.
network command character	This manual uses the default character (%).
numbers	All numbers are decimal unless otherwise noted.
UPPERCASE	Uppercase is used to depict names of commands, parameters (in some cases, called attributes) and key names. For example, the RETURN key is the carriage-return key.

Related CDCNET Manuals



 Available on line only.

Additional Related Manuals

The following Control Data publications provide further information about some of the topics described in this manual.

Title	Publication Number
CDCNET TCP/IP	60000214
CDCNET Configuration Guide	60461550
CDCNET Batch Device User's Guide	60463863
CDCNET Commands Reference Manual	60000414
Network Configuration Utility (NETCU) Online Manual	60000268
CYBIL File Management	60464114

Title	Publication Number
Network Products, Network Access Method, Version 1, Host Application Programming Reference Manual	60499500
NOS Version 2 Reference Set Volume 3, System Commands	60459680
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS/VE System Usage	60464014
Network Job Entry Facility Installation Manual	15190118

Ordering Manuals

Control Data manuals are available through Control Data sales offices or through:

Control Data
Literature and Distribution Services
308 North Dale Street
St. Paul, MN 55103-2495

Submitting Comments

The last page of this manual is a comment sheet. Please use it to give us your opinion of the manual's usability, to suggest specific improvements, and to report technical or typographical errors. If the comment sheet has already been used, you can mail your comments to:

Control Data
Technical Publications ARH219
4201 Lexington Avenue North
St. Paul, MN 55126-6198

Please indicate if you would like a written response.

You can also submit your comments through SOLVER, an online facility for reporting problems. To submit a documentation comment through SOLVER, do the following:

1. Select option 3 from the main SOLVER menu:

   ```
   Report a new problem or change an existing PSR.
   ```

2. Respond to the prompts for site-specific information.

3. Select option 4 from the new menu:

   ```
   Write a comment about a manual.
   ```

4. Respond to the prompts.

(For information on accessing SOLVER, contact your local Control Data sales office.)

Introduction 1

Introduction 1

The Control Data Distributed Communications Network (CDCNET) connects your terminal to the service you want to use. This chapter describes:

- Accessing and leaving a service
- Using CDCNET
- Understanding configuration options

Accessing and Leaving a Service

CDCNET (also called the network) provides a path called a connection that enables you to access and communicate with your computing service. The following instructions describe the four actions you must take to use your service.

1. Connect to CDCNET
2. Connect to a Service
3. Disconnect from a Service
4. Disconnect from CDCNET

Connecting to CDCNET

To connect your terminal to CDCNET:

1. Have your computer identification information available. This includes phone numbers, service names, terminal model names, and validation information that you have received from your management or your site administrator.

2. Turn your terminal on. If you have any questions about how it operates, consult your terminal documentation or a terminal resource person at your site.

3. If your terminal is connected directly to your network (dedicated), you are ready for the next step. If not, follow your local procedure for connecting a terminal through a modem or data set to the network (switched). This may involve manually dialing a phone number you have been given.

4. Press your RETURN key (or its equivalent) twice.[1] CDCNET responds with a message that looks something like this:

```
9600 bps ASCII, parity:  odd

Copyright Control Data Corporation, 1985, 1989
DI System Name is 080025100081, TDI_A1
Terminal Name is 5300000000, $CONSOLE_100081_5300000000
You may enter CDCNET commands.
```

   This message identifies the network equipment (DI System Name) and your terminal (Terminal Name). Make a note of this information, as it can be useful in solving communications problems. The site administrator assigns the *logical* names for this equipment; these are the names that appear after the commas in the display. In the example above, the logical DI name is TDI_A1 and the logical terminal name is $CONSOLE_100081_5300000000.

   NOTE

   If you do not receive any messages or receive only garbled (unreadable) messages, consult your site administrator.

   You are now connected to CDCNET by way of the $NET connection. This is not a service connection, but a special connection that allows you to communicate directly with CDCNET by entering CDCNET terminal user commands.

5. If you are working at a facility where network security is a concern, enter the secure access sequence, described in chapter 6, to ensure that you are connected directly to CDCNET.

1. If you are using APL, press an APL right parenthesis [)], instead of pressing the RETURN key a second time. For the typewriter-pairing ASCII coded character set with APL print, this parenthesis corresponds to an ASCII asterisk [*]. For the bit-pairing ASCII coded character set with APL print, the parenthesis corresponds to an ASCII quotation mark ["].

Connecting to a Service

To connect to a service:

1. Some sites may have a special procedure that automatically configures your terminal and/or accesses your computing service. If your management or site administrator gave you the name of a procedure, execute it by entering a DO command followed by the procedure name.

   ```
   DO procedure name
   ```

2. If your procedure (step 1) automatically accesses your service, skip to step 3. Otherwise, using the service name provided by your site administrator, enter the following CDCNET command to access your service.

   ```
   CREC service name
   ```

   CREC is the abbreviation for the CREATE_CONNECTION command.

3. Enter any validation information (such as username and password) required by the service, and then begin your processing.

Disconnecting from a Service

When you are ready to stop processing, log off of the service you are connected to by entering the appropriate service-specific command. For example, enter LOGOUT to terminate your NOS/VE session.

Logging out of the service causes CDCNET to delete the associated connection and display the following prompt:

```
You may enter CDCNET commands.
```

If you have other service connections on hold when you log off, CDCNET displays these connections before displaying its prompt. This service connection display is formatted as follows:

```
Connection_Name               Service_Name

  $A                            ABC
  EDIT                          XYZ
```

Disconnecting from CDCNET

To disconnect from CDCNET, do one of the following:

- If your terminal is using a switched line, hang up the phone.
- If your terminal is using a dedicated line, turn off your terminal.

Using CDCNET

When you are working at a terminal connected to a service, you can normally begin processing tasks without further regard to CDCNET connections and terminal characteristics. However, if you are working as a customer service analyst, writing application programs, or using a microcomputer, you may need to know more about CDCNET features.

When You Need To Know More About CDCNET

Customer service analysts frequently help users resolve problems with terminal characteristics. Some terminals do require additional changes when their characteristics differ from the default settings established for the site's network. Sometimes users need help in changing their terminal characteristics to benefit from different network capabilities.

If you are writing programs or applications, you may need to know more about changing terminal characteristics so that users can successfully work with your products.

If you are using a microcomputer, you may want to tailor the network to your specific needs. For example, file transfers may require you to use different network options.

Although this manual specifically describes how to control network capabilities, you may want to be aware of other options to ensure efficient and correct operation. Changes in the overall presentation of data to the service and to the user can be made from the service, network, and terminals, especially microcomputers, as shown in the following table.

Area	Example
Service	On some services, you can use commands to change the terminal characteristics called terminal attributes. For example, NOS/VE services support the following: CHANGE_TERMINAL_ATTRIBUTE SET_TERMINAL_ATTRIBUTE IFP$CHANGE_TERMINAL_ATTRIBUTE (See appendixes D and E.)
CDCNET	You can also manipulate terminal attributes with the CHANGE_TERMINAL_ATTRIBUTE command.
Microcomputer	As with most terminals, you may change the physical attributes of microcomputers by resetting switches or software parameters. Depending upon the intelligence of the microcomputer, you may also translate data (filter or suppress it) via communication packages (such as CONNECT).

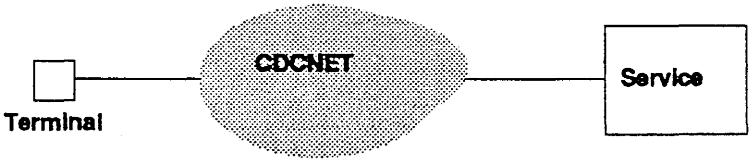
Your Network

This section discusses the following network components:

- Terminals
- Other networks
- Slave devices

Terminals

As mentioned earlier, the word **terminal** in this manual means any interactive device used to access a service through CDCNET. It may be a microcomputer, batch console, video-display terminal, hardcopy terminal, display station, or workstation that you can use to conduct an interactive dialog with your service. It is not a batch device with only an input or output mechanism. (See the CDCNET Batch Device User Guide for batch information.)



Using Other Networks

Your site may use a local area network (LAN) or a public data network (PDN) between your terminal and CDCNET. In other words, your terminal may go through another network to connect with CDCNET and subsequently access a service. If so, observe these guidelines to expedite CDCNET processing.

- Be aware of any parameter setting on the LAN or PDN that may affect your CDCNET processing.

- Follow your local procedure for accessing and using the LAN or PDN.

- Work with your site administrator to solve any difficulties you may encounter.

Using Slave Devices

Although terminals can have multiple input or output mechanisms on a connection at any time, CDCNET is aware of only one. Such mechanisms operate in series and are called **slave devices**. For example, microcomputer users frequently use printers and disk drives as slave devices. In these cases, the network does not distinguish the input of a microcomputer's disk unit from keyboard input. Similarly, the network treats output to a hardcopy printer the same as display output.

CDCNET Features

CDCNET capabilities, designed to simplify control of your processing, are provided by four types of features.

Type of Feature	Description
Commands	The network recognizes commands that control the way it operates your terminal and service. (See chapter 2.)
Terminal attributes	The network supports terminal attributes that you may display and change. These attributes set characteristics that affect how your terminal communicates with the network, and that apply to all of your connections. (See chapter 3.)
Connection attributes	The network also supports connection attributes that you can display and change. These regulate certain aspects of *a connection* between your terminal and the service. Each connection has its own set of connection attributes.[2] (See chapter 4.)
Procedures	Your site administrator uses procedures in configuring your network. One of these procedures may establish special terminal-attribute and connection-attribute defaults for your terminal. (See the section on Understanding Configuration Options next in this chapter.)

2. $NET connection attributes (IEM=NORMAL, IOM=SOLICITED) cannot be displayed or changed.

Understanding Configuration Options

Certain operating characteristics can be explicitly set by the site administrator when configuring the network. Some of the characteristics established by the DEFINE_LINE command (used by your site administrator) directly affect how you access your service and how your terminal operates. These characteristics are controlled strictly by the site administrator and, in most cases, you cannot override them. Specifically, the site administrator sets parameters on the DEFINE_LINE command that determine the following:

- Terminal interface program (protocol)
- Line type
- Line speed
- Automatic recognition
- Connect timeout
- Disconnect timeout
- Terminal definition procedure
- Terminal user procedure
- Connection limit
- EIA flow control
- Parity

DEFINE_LINE is described in the CDCNET Configuration Guide.

The following sections describe the configuration options.

Terminal Interface Program (Protocol)

CDCNET supports a number of communication protocols. Each supported protocol has an associated terminal interface program (TIP) that allows users of that protocol to interface with CDCNET. CDCNET provides the following terminal interface programs, supporting their respective protocols.

TIP Name	Supported Protocol	Description
ASYNCTIP	Asynchronous	ASYNCTIP supports the use of teletype-compatible terminals connected through dedicated or switched asynchronous communication lines via the Asynchronous protocol. The TIP supports one device per communication line, and expects each device to have both an input and output mechanism (terminal), or an output mechanism (printer) only.
HASPTIP	HASP	HASPTIP supports the use of consoles that are attached to HASP workstations through the HASP protocol. This TIP enables your communication line and allows you to perform limited data editing and cursor positioning. (See the CDCNET Batch Device User Guide for information on card readers, line printers, card punches, and plotters.)

TIP Name	Supported Protocol	Description
MODE4TIP	Mode 4	MODE4TIP lets users of Mode 4 terminals connect to services through CDCNET via the Mode 4 protocol. There are two versions of the protocol: Mode 4A and Mode 4C. Mode 4A supports a group of devices, such as a console, line printer, and card reader. Mode 4C supports several clusters of consoles and line printers. See appendix I for more information.
URITIP	Unit Record Interface (URI)	URITIP supports character and line printers using the URI protocol. The protocol is managed by the network, and user action is not necessary. See the CDCNET Batch Device User Guide for further information.

TIP Name	Supported Protocol	Description
XPCTIP	X.PC	XPCTIP lets personal computer users access CDCNET services over switched or dedicated asynchronous lines using the packet-switching communications techniques of the X.PC protocol. The TIP supports up to 15 simultaneous connections over a single link, providing the personal computer user with powerful, advanced capabilities, such as windowing and error-free data transfers. See appendix H for more information.
X25_ASYNCTIP	X.25 Asynchronous	X.25_ASYNCTIP supports asynchronous terminals that are connected to CDCNET through an X.29 PAD (Packet Assembler/Disassembler) using the X.25 Asynchronous protocol. See appendix G for more information. The TIP also supports use of the X.PC protocol over X.25 asynchronous lines.
BSC3270TIP	3270 Binary Synchronous Communications	BSC3270TIP enables users of IBM 3270 Information Display Systems (terminals) to connect to CDCNET and use NOS and NOS/VE services via the 3270 Binary Synchronous Communications protocol. See appendix F for more information.

TIP Name	Supported Protocol	Description
SNA3270_TIP	3270 SNA Communications	SNA3270_TIP enables IBM 3270 Information Display Systems (terminals), connected to an SNA network, to access CDCNET through the 3270 SNA Communications protocol and use NOS and NOS/VE services. See appendix J for more information.

The protocol you use determines which terminal attributes and connection attributes are available to you in the associated TIP. Each protocol has its own subset of attributes and defaults. For example, the Asynchronous protocol uses all of the available attributes, while the HASP and 3270 Binary Synchronous Communications protocols use small subsets of ASYNCTIP attributes.

In addition, the network has provisions for site-defined TIPs supporting protocols currently not supported by CDCNET TIPs. If a site-defined TIP is used at your site, your site administrator is responsible for documenting it.

Line Type

The network recognizes two types of communication lines.

Type of Line	Description
Switched	A switched line typically requires a dial-in procedure and modem.
Dedicated	A dedicated line is usually connected directly to the network.

Line Speed

The site administrator defines the speed of a communication line in bits per second (bps) or, alternatively, the site administrator can enable automatic recognition of line speed. The network accepts the following line speeds for the Asynchronous, HASP, Mode 4, X.PC, and 3270 Binary Synchronous Communications protocols.[3]

```
                     50
                     75
                    110  Asynchronous and X.PC Protocols
                    150
                    300
                    600
                   1200
                   2400   3270 Binary Synchronous
Mode 4 Protocol    4800   Communications Protocol
                   9600
                  19200
                  38400
                  48000
  HASP Protocol   56000
                  64000
```

When transmitting data to CDCNET, terminals using the asynchronous protocol send a signal, called a stop bit, at the end of each character to prepare the network for the following character. The network assumes that terminals operating at 110 bits per second or slower use two stop bits, and terminals with speeds of 150 bits per second or higher use one. You may have to change a setting on your terminal to match the network's expectation.

3. Trunk speeds for X.25 Asynchronous and 3270 SNA Communications protocols are not described here, because terminals using these protocols are not directly connected to CDCNET.

Automatic Recognition

If you are using the asynchronous protocol, your site administrator may request CDCNET to determine certain things about your terminal when it joins the network. The administrator can specify automatic recognition of any of the following:

- Line speed only
- Line speed and coded character set
- Line speed, coded character set, and parity

If the site administrator omits this parameter, the network does not perform any automatic recognition. However, it does force the following settings when automatic recognition of coded character set and parity is not requested.

CODE_SET=ASCII
PARITY=EVEN

On a switched line, you have 90 seconds to complete automatic recognition before the network disconnects your terminal.

Terminals reaching CDCNET through another network usually do not use the CDCNET automatic recognition process. However, the other network may have a corresponding process.

Connect Timeout

After you access CDCNET, you have a specified amount of time to create a connection to a service. If you do not establish one within that time, the network disconnects the communication line.

Your site administrator can specify any limit from 20 through 1000 seconds. If the administrator omits this setting, the network sets the connect timeout at 2 minutes (120 seconds) for a switched line and INFINITE for a dedicated line.

If you are also using automatic recognition, the network allows 90 seconds for that process, plus the time specified on this parameter, before disconnecting the communication line.

Disconnect Timeout

When you are using the $NET connection and do not have any other connection, you have a specified amount of time to establish a new connection to a service. If you do not enter a CREATE_CONNECTION command within that time, the network disconnects the communication line.

Your site administrator can specify any limit from 20 through 1000 seconds. If the administrator omits this setting, the network sets the disconnect timeout at 2 minutes (120 seconds) for a switched line and INFINITE for a dedicated line.

Terminal Definition Procedure (TDP)

A site administrator can configure the network for the type of equipment on a communication line by creating a terminal definition procedure (TDP) that the network associates with the communication line. If the site administrator creates both a TDP and a terminal user procedure (TUP) (described next) on the DEFINE_LINE command, the network executes only the TDP. However, if the site administrator defines a TUP on a DEFINE_TERMINAL_DEVICE command within the TDP, the network executes both the TDP and that TUP.

Terminal User Procedure (TUP)

If the site administrator wants to change your access procedure or establish individual attribute defaults for your terminal, the administrator can create a terminal user procedure (TUP). When you access the network, it automatically executes this procedure. A TUP may contain any terminal user commands described in this manual, except ACTIVATE_AUTO_RECOGNITION.

The network executes both the TUP and the TDP if the site administrator has defined the TUP on a DEFINE_TERMINAL_DEVICE command within the TDP. Otherwise, the network executes only the TDP.

For more information on TDP and TUP, see the CDCNET Configuration Guide.

Connection Limit

CDCNET allows you to create and use a number of connections to link your terminal to computing services. The connection you are currently using is called your working connection. When you initially access CDCNET, your working connection is the default connection, $NET.

You can have multiple connections to one or more services. The site administrator can configure the maximum number of connections. If the administrator omits this parameter, the network allows each terminal four connections. For the X.PC protocol, a virtual circuit with no service connections is counted as having one connection for the purposes of checking against this limit.

EIA Flow Control

The site administrator specifies whether or not you can use EIA flow control to control the flow of input. (See chapter 5 for additional information on flow control.) If the administrator omits this parameter on the DEFINE_LINE command, the network does not use EIA flow control for input. However, the network always responds to EIA flow control received from a terminal for output.

Parity

The site administrator can specify parity for data received and transmitted on this line if the administrator has not selected automatic recognition of parity. The setting of the DEFINE_LINE parameter is also the setting of the PARITY terminal attribute when automatic recognition of parity is not specified. If the site administrator does not select one of the following settings, the network uses EVEN.

EVEN
ODD
MARK
ZERO
NONE

(See Connecting to CDCNET earlier in this chapter and the description of the PARITY terminal attribute in chapter 3 for more information.)

CDCNET Terminal Interface Commands 2

CDCNET Terminal Interface Commands 2

This chapter describes CDCNET terminal interface commands. Most of these commands can be entered either from an interactive session or from within a terminal procedure (exceptions are noted in the command descriptions). For information about constructing and using terminal procedures, refer to the CDCNET Configuration Guide.

CDCNET terminal user commands can be entered at any time, whether you are connected to the default CDCNET connection (called $NET) or to a service. The way you enter a CDCNET command on a service connection depends on whether the INPUT_EDITING_MODE connection attribute is set to NORMAL or TRANSPARENT. The differences are summarized in the following table:

Type of Connection	How to Enter a CDCNET Terminal User Command
$NET	You can enter the CDCNET terminal user commands described in this chapter *without prefixing them with the network command character.*
Service Connection, Normal Mode	You can enter the CDCNET terminal user commands described in this chapter *by prefixing them with the network command character.* CDCNET looks for the network command character on normal input and tries to execute what follows as a user interrupt or command.
Service Connection, Transparent Mode	You can enter the CDCNET terminal user commands described in this chapter only after using one of the user break sequences described in chapter 6.

Command formats and examples in this chapter assume you are using a service connection that is in normal input editing mode. The default network command character, a percent sign (%), is used in all formats and most examples. You can change the network command character with the CHANGE_TERMINAL_ATTRIBUTE command described later in this chapter.

Command Conventions

CDCNET commands use NOS/VE System Command Language (SCL) conventions. The following sections describe the general structure of these commands, as used interactively and in procedures. The following conventions apply to all CDCNET commands.

- Each command must appear on a separate line.
- A command can contain as many as 256 characters.
- With the exception of strings, uppercase and lowercase letters are interchangeable in commands. (See the discussion of parameter values later in this chapter.)

Command Format

CDCNET command names consist of a verb that describes the operation, followed by an object that represents the target of the operation. The verb and object are concatenated by an underline character (_), which is part of the name and cannot be omitted or replaced by a space. For example:

```
CREATE_CONNECTION
```

In this example, the underline connects the verb (CREATE) to its object (CONNECTION).

CDCNET commands entered on a service connection operating in NORMAL input editing mode must be preceded by the network command character. This manual uses the default network command character for the Asynchronous protocol, a percent sign (%). For example:

```
%CREATE_CONNECTION
```

When you enter a command from the $NET connection, you can omit the NETWORK_COMMAND_CHARACTER because this connection handles only CDCNET communications. It does not communicate with a service.

Like command names, parameter names are concatenated if they have more than one word. In the following example, the underline joins the two parts of the parameter name (SERVICE_NAME).

```
%CREATE_CONNECTION SERVICE_NAME=service name
```

After you enter a command and its parameters, press the RETURN key (or equivalent) to execute the command.

NOTE

Separate the command name and each parameter with spaces or commas. For example:

```
%CREATE_CONNECTION SERVICE_NAME=service name OUTPUT_ACTION=output action

%CREATE_CONNECTION,SERVICE_NAME=service name,OUTPUT_ACTION=output action
```

Abbreviations

When you enter commands and parameters, you can use their spelled-out names, their abbreviations, or a combination of the two. For example:

```
CREATE_CONNECTION SN=service_name OUTPUT_ACTION=output_action
```

CDCNET commands are abbreviated by taking the first three characters from the verb portion of the command name and concatenating them with the first character from each subsequent word in the command name (CREATE_CONNECTION is abbreviated CREC). Generally, the first character from each word in a parameter name is used to abbreviate parameters (SERVICE_NAME is abbreviated SN).

If a command has a singular and plural form, the abbreviation applies to both. For example, DISTA is the abbreviation for DISPLAY_TERMINAL_ATTRIBUTE and DISPLAY_TERMINAL_ATTRIBUTES.

Parameter Position

When you enter CDCNET commands at a terminal, you can specify position-independent or position-dependent parameters.

Position Independent

Position-independent parameters consist of a parameter name or abbreviation, an equal sign, and a value. That is,

```
PARAMETER_NAME=value
```

You can place these parameters in any order. For example:

```
%CREC OA=output action SN=service name
```

Position Dependent

Position-dependent parameters contain only the value, not the parameter name or equal sign. You specify them in the order shown in the command description. If you omit a parameter, separate the parameters with commas and insert an extra comma for the missing parameter. In the following CREATE_CONNECTION examples, the extra comma skips over the second parameter, the CONNECTION_NAME parameter.

```
%CREATE_CONNECTION service_name,,output_action

%CREC service_name,,output_action
```

Parameter Values

Network commands define the kinds of values you can assign parameters, and the ranges these values can have.

CDCNET commands use the following kinds of System Command Language (SCL) parameter values:

- Name
- String
- Integer
- Boolean
- Keyword

In addition, the commands use the following CDCNET-defined parameter values.

- Character
- Sequence
- List
- Message

These parameter value types are described in the following sections.

Command Summary

Table 2-1 summarizes the commands described in this chapter.

Table 2-1. Using CDCNET Commands

Command	Purpose
ACTIVATE_AUTO_RECOGNITION	Resets your terminal's line speed, coded character set, and/or parity.
ACTIVATE_X_PERSONAL_ COMPUTER	Switches a microcomputer connection from asynchronous protocol to X.PC protocol.
CHANGE_CONNECTION_ ATTRIBUTE	Changes any of the attributes the network uses to control your working connection.
CHANGE_TERMINAL_ATTRIBUTE	Changes any of the attributes the network uses to control your terminal device. These attributes apply to all active connections.
CHANGE_WORKING_CONNECTION	Changes your working connection from one existing connection to another.
CREATE_CONNECTION	Creates a new CDCNET connection.
DELETE_CONNECTION	Deletes an existing CDCNET connection.
DISPLAY_COMMAND_ INFORMATION	Displays summary information about any of the CDCNET terminal user commands.

(Continued)

Table 2-1. Using CDCNET Commands *(Continued)*

Command	Purpose
DISPLAY_COMMAND_LIST	Displays a list of the CDCNET terminal user commands.
DISPLAY_CONNECTION_ATTRIBUTES	Displays the connection attributes associated with the working connection.
DISPLAY_CONNECTIONS	Displays summary information about existing CDCNET connections.
DISPLAY_SERVICE	Displays a list of some or all of the computing services available on the network.
DISPLAY_TERMINAL_ATTRIBUTE	Displays the attributes associated with your terminal device.
DO	Executes a terminal user or terminal definition procedure.
REQUEST_NETWORK_OPERATOR	Sends a message to the network operator.

The following pages describe the CDCNET terminal user commands, in alphabetical order. These commands are presented in a quick-reference format that includes purpose, format, parameters, remarks, and examples, when applicable. Each description begins with a command name followed by its abbreviation, enclosed in parentheses. Under the parameter descriptions, each parameter name is followed by its abbreviation, in parentheses.

ACTIVATE_AUTO_RECOGNITION (ACTAR)

Purpose

Directs the network to test your terminal's line speed, coded character set, and/or parity, and to adjust the line's settings accordingly. This command can only be entered interactively; it cannot be used in a terminal user procedure (TUP).

NOTE

The level of automatic recognition varies, depending upon how the line was defined. The possibilities are as follows:

1. Automatic recognition of line speed, code set and parity
2. Automatic recognition of line speed and code set
3. Automatic recognition of line speed only
4. No automatic recognition

Ask your site administrator what level of automatic recognition your line uses.

Format

%ACTIVATE_AUTO_RECOGNITION

Remarks

- If the site administrator defined your line for automatic recognition of speed, coded character set, and parity, here is what happens when you use the ACTAR command:

  1. After you enter the ACTAR command (complete with a single RETURN), the network is ready for you to change your terminal's line speed, and/or code set and parity.

  2. After you make the desired changes to your terminal's settings, press the RETURN key again. The network uses this character (usually CR) to detect line speed, and returns two line-feed characters in the original PARITY setting (some terminals may report parity errors on these two characters).

3. Now change your terminal's code set and parity settings (if you haven't already done so).

4. If you are using the ASCII coded character set, press the RETURN key once more. The network uses this character to determine coded character set and parity, and it changes its settings of the CODE_SET and PARITY terminal attributes to match your terminal's new settings.

   If you are using APL, press an APL right parenthesis [)] instead of pressing the RETURN key again. For the typewriter-pairing ASCII coded character set with APL print, this parenthesis is an ASCII asterisk (*). For the bit-pairing ASCII coded character set with APL print, the parenthesis is an ASCII quotation mark (").

5. The network returns a message that indicates your terminal's new line speed, coded character set and parity. Depending upon the line speed, the network issues NUL characters before displaying the message, as follows:

Line Speed	Number of NUL Characters
110 through 600 bps	2 NUL characters
1200 through 9600 bps	1 NUL character
19200 through 38400 bps	None

   The network inserts 2 NUL characters after each message.

- If the site administrator defined your line for automatic recognition of your terminal's parity, the network sets the PARITY terminal attribute to the following:

Terminal's Parity	Terminal Attribute PARITY Setting
EVEN	EVEN
MARK	EVEN
ODD	ODD
ZERO	ODD
NONE	ODD

- X.PC line speed can be automatically recognized. However, character set and parity are ignored during automatic recognition. The X.PC character set is always ASCII, and parity is set by the X.PC protocol.

- Devices such as PCs, which use procedures to perform auto-recognition, should use a 2-second delay between the first and second carriage returns of the auto-recognition sequence.

- See Automatic Recognition Difficulties, chapter 8, for additional information about automatic recognition.

Examples

If you use the ASCII coded character set and your line was defined for automatic recognition, you can change the line speed to 9600 bits per second (bps) in the following three steps:

1. Enter the following command (complete with a single carriage return):

```
%ACTAR
```

2. Switch the terminal's line speed to 9600 bps (consult your terminal's user guide).

3. Press the RETURN key twice.

The network responds with a message that identifies your terminal speed, coded character set, and parity. For example:

```
9600 bps ASCII, parity: even
```

ACTIVATE_X_PERSONAL_COMPUTER (ACTXPC)

Purpose

Switches a microcomputer's connection to operate in the X.PC protocol mode (it is not necessary to use this command if your microcomputer's CDCNET line is configured specifically for the X.PC protocol). When the connection is switched to the X.PC protocol mode, the microcomputer is directed to start the X.PC protocol. See appendix H for a description of the CDCNET support of the X.PC protocol.

Format

%ACTIVATE_X_PERSONAL_COMPUTER
TIMEOUT=integer
OPTIMIZE_PACKETS=boolean

TIMEOUT (T)

Specifies the time limit during which the network retransmits unacknowledged packets. Default is 4 seconds.

OPTIMIZE_PACKETS (OP)

Specifies whether or not the network must reduce the size of the X.PC packet to a more optimum size for X.25 packets. Default is FALSE.

Remarks

- Your microcomputer must be connected to CDCNET using the asynchronous or X.25 asynchronous protocol. If connected using the asynchronous protocol, the DI must be configured for the X.PC protocol. If connected to the X.25 asynchronous protocol, the DI does not have to be configured for the X.PC protocol. The microcomputer software must switch to X.PC mode upon notification that the network is activating the X.PC protocol.

- When the network receives this command, it verifies the correct protocol is currently in use, and that the coded character set is ASCII. The X.PC protocol is then started for that line. X.PC informs you that the command completed successfully by sending the following message:

```
X.PC protocol being activated.
```

- If your microcomputer does not start the X.PC protocol within 1.5 minutes of receiving this message, the microcomputer is disconnected. If connected using the asynchronous protocol, the line is then disconnected. If connected using the X.25 asynchronous protocol, the virtual circuit is then cleared.

- You do not need to use the ACTXPC command if your microcomputer is directly configured to the X.PC protocol (the site administrator defined your line for use with X.PC), or if your microcomputer's data communications software issues this command automatically. Ask your site administrator how your data communications line is configured and refer to your microcomputer data communications software documentation to understand how X.PC is activated on your microcomputer system.

CHANGE_CONNECTION_ATTRIBUTE (CHACA)

Purpose Specifies changes you want made to your current connection attributes.

Format %CHANGE_CONNECTION_ATTRIBUTE

ATTENTION_CHARACTER_ACTION=integer
ATTENTION_CHARACTER_ENABLE=keyword
BREAK_KEY_ACTION=integer
ECHO_ENABLE=keyword
INPUT_BLOCK_SIZE=integer
INPUT_EDITING_MODE=keyword
INPUT_FLOW_CONTROL_ENABLE=keyword
INPUT_OUTPUT_MODE=keyword
OUTPUT_FLOW_CONTROL_ENABLE=keyword
PARITY_ENABLE=keyword
PARTIAL_CHARACTER_FORWARDING=keyword
STORE_BACKSPACE_CHARACTER=keyword
STORE_NULS_DELS=keyword
TRANSPARENT_CHARACTER_MODE=keyword
TRANSPARENT_FORWARD_CHARACTER=list
TRANSPARENT_LENGTH_MODE=keyword
TRANSPARENT_MESSAGE_LENGTH=integer
TRANSPARENT_PROTOCOL_MODE=keyword
TRANSPARENT_TERMINATE_CHARACTER=list
TRANSPARENT_TIMEOUT_INTERVAL=integer
TRANSPARENT_TIMEOUT_MODE=keyword

Parameters Each CHACA parameter has a corresponding connection attribute description in chapter 4.

Remarks

- The CHACA command does not apply when your working connection is $NET.
- Use of a DO command may change your connection attributes if the procedure contains a CHACA command. You should be aware of the effects of a procedure before you execute it.
- When writing application programs, you must change connection attributes by using commands available to your application via the service you are using.

Examples The following CHACA command changes the values of STORE_BACKSPACE_CHARACTER and STORE_NULS_DELS from OFF to ON.

```
%CHACA SBC=ON SND=ON
```

The network makes the changes and responds with a message:

```
Attributes changed.
```

CHANGE_TERMINAL_ATTRIBUTE (CHATA)

Purpose Specifies changes you want to make to your current terminal attributes.

Format %CHANGE_TERMINAL_ATTRIBUTE

ATTENTION_CHARACTER=character
BACKSPACE_CHARACTER=character
BEGIN_LINE_CHARACTER=character
CANCEL_LINE_CHARACTER=character
CARRIAGE_RETURN_DELAY=integer
CARRIAGE_RETURN_SEQUENCE=sequence
CHARACTER_FLOW_CONTROL=keyword
CODE_SET=name or keyword
ECHOPLEX=keyword
END_LINE_CHARACTER=character
END_LINE_POSITIONING=keyword
END_OUTPUT_SEQUENCE=sequence
END_PAGE_ACTION=keyword
END_PARTIAL_CHARACTER=character
END_PARTIAL_POSITIONING=keyword
FOLD_LINE=keyword
FORM_FEED_DELAY=integer
FORM_FEED_SEQUENCE=sequence
HOLD_PAGE=keyword
HOLD_PAGE_OVER=keyword
LINE_FEED_DELAY=integer
LINE_FEED_SEQUENCE=sequence
NETWORK_COMMAND_CHARACTER=character
PAGE_LENGTH=integer
PAGE_WIDTH=integer
PARITY=keyword
RESPONSE_ACTION=keyword
STATUS_ACTION=keyword
TERMINAL_MODEL=name
BACKSPACE_WINDOW=integer
CONTROL_CODE_REPLACEMENT=list
FUNCTION_KEY_CLASS=name

Parameters Each CHANGE_TERMINAL_ATTRIBUTE parameter has a corresponding terminal attribute description in chapter 3.

Remarks

- Use of a DO command may change your terminal attributes if the procedure contains a CHANGE_TERMINAL_ATTRIBUTE command. You should be aware of the effects of a procedure before you execute it.
- You can change a terminal attribute with the CHANGE_TERMINAL_ATTRIBUTE command even if the protocol your terminal uses does not support it. The changed value for the nonsupported terminal attribute, however, has no effect and is not validated by your terminal's protocol.

Examples

By specifying a series, you can change several terminal attributes using a single command. The following CHANGE_TERMINAL_ATTRIBUTE command changes the values of PAGE_LENGTH and HOLD_PAGE. These changes extend the page length to 30 lines and display output one page at a time.

```
%CHATA PL=30 HP=ON
```

You can also change a single attribute.

```
%CHATA PL=0
```

In this example, the CHANGE_TERMINAL_ATTRIBUTE command changes PAGE_LENGTH to 0 so that your output is continuous, without page breaks.

CHANGE_WORKING_CONNECTION (CHAWC)

Purpose Makes an alternate existing connection the working connection, without deleting the connection you are leaving. The working connection is the connection your terminal currently uses to exchange information on the network.

Format %CHANGE_WORKING_CONNECTION
CONNECTION_NAME=connection name
OUTPUT_ACTION=output action

Parameters *CONNECTION_NAME (CN)*

Identifies the new working connection. If you do not specify a new working connection, CHAWC makes the $NET connection your working connection.

OUTPUT_ACTION (OA)

Specifies one of the following entries:

DISCARD (D)	Discards new output from the connection you are leaving.
HOLD (H)	Holds new output from the connection you are leaving. When you return to this connection, the output is displayed.

If you do not enter an output action, the network uses HOLD.

Remarks

- This command lets you use multiple connections simultaneously. You can move from one connection to another to obtain job output, display status reports, edit, and perform other activities just as you can between terminals.
- Using the CHANGE_WORKING_CONNECTION command to switch your working connection does not delete the connection you are leaving.
- You cannot select a connection that has a connection status of PENDING as your working connection.

Examples

The following command changes your working connection to an existing connection named EDIT, on a service named ABC, and holds output from the connection you are leaving:

```
%CHAWC EDIT
```

The network responds with the following message:

```
Working connection changed to EDIT, service name ABC.
```

You can retrieve the output from the previous working connection by changing the current (working) connection to the previous connection.

The following command changes your connection to $NET, which is not connected to any service:

```
%CHAWC
```

CDCNET responds with a display of all existing service connections, and a prompt for CDCNET terminal user commands. The display is formatted as follows:

```
Connection_Name              Service_Name

  $A                           ABC
  EDIT                         XYZ

You may enter CDCNET commands.
```

CREATE_CONNECTION (CREC)

Purpose

Creates a connection to the service you select and automatically makes it your working connection. This command is necessary to access a service through CDCNET.

Format

%CREATE_CONNECTION
  SERVICE_NAME=service name
  CONNECTION_NAME=connection name
  OUTPUT_ACTION=output action
  WAIT=boolean
  SERVICE_DATA=string

Parameters

SERVICE_NAME (SN)

Specifies the name of the service or gateway to which you wish to connect. See your site administrator for the names of services and gateways on your network.

CONNECTION_NAME (CN)

Assigns a unique name to the connection. This name can be 1 through 31 characters long. It must begin with a non-numeric, but can contain a combination of letters, numerics, and special characters (_, $,[1] #, @). If you do not enter a connection name, the network assigns a system-reserved name beginning with a dollar sign ($).

OUTPUT_ACTION (OA)

Specifies one of the following:

DISCARD (D)	Discards new output from the working connection you are leaving.
HOLD (H)	Holds new output from the connection you are leaving. When you return to this connection, the output is displayed.

Default is HOLD.

1. The dollar sign ($) cannot be used to begin a user-specified connection name, but can be used anywhere else in the name.

WAIT (W)

Specifies one of the following:

YES	If the service you requested is unavailable, a connection is created and put in a state where it waits for the service to become available. If and when the service becomes available, you receive a message indicating the requested connection has been completed.
NO	If the service is unavailable, the connection request is terminated, and you receive a message indicating the requested service is unavailable.

The default is NO.

SERVICE_DATA (SD)

Specifies a string of 1 to 63 characters that is interpreted or required by the service to which you are requesting the connection.

Remarks

- You can create more than one connection to the same service.
- If creating another connection would cause you to exceed the connection limit your site administrator set, the network issues an error message instead of connecting your terminal to the service. Use the DISPLAY_CONNECTION command to check your connection limit.
- Use the CHANGE_WORKING_CONNECTION command to alternate between existing CDCNET connections.

Examples

The following command, in response to the CDCNET prompt, creates an initial service connection to a service named ABC:

```
CREC SN=ABC
```

The network command character is not required with this command because the working connection is still $NET. The following message is displayed prior to any banner message supplied by the ABC service:

```
Connection $A created.
```

While connected to service ABC, a second connection is created by prefixing the CREC command with the network command character (%). For example, the following command creates a connection to service XYZ, and assigns this connection the name EDIT.

```
%CREC XYZ EDIT
```

The following message is displayed prior to any banner message supplied by service XYZ:

```
Connection EDIT created.
```

The following is an example of how the SERVICE_DATA parameter is used in the CREC command to create a CDCNET connection to a User TELNET gateway named TELNET. After the connection is made to the User TELNET gateway, the gateway uses the information provided in the SERVICE_DATA parameter to establish a TELNET connection to a remote TCP/IP host.

```
CREC,TELNET, SD='128.2.53.7'
```

The following example shows CREC command used with the WAIT parameter specified to YES.

```
CREC ARHNOS W=YES
```

The following message is displayed if ARHNOS is unavailable.

```
Searching for ARHNOS.
     Connection $A pending.
```

DELETE_CONNECTION (DELC)

Purpose

Deletes the specified connection. If you delete a connection other than the current working connection, the session on your current working connection continues. If you delete the current working connection, you are connected to the $NET connection and you receive a list of all other connections currently on HOLD. This display also occurs when you log off the service and other connections are currently on HOLD. You can then create a new connection (using CREC) or change to another existing connection (using CHAWC).

Format

%DELETE_CONNECTION
CONNECTION_NAME=$NET *or connection name*

Parameters

CONNECTION_NAME (CN)

Specifies the name of the connection to be deleted. This parameter is required only when you want to delete the $NET connection. The parameter value defaults to the current working connection when entered on a connection other than the $NET connection.

Remarks

- This command disconnects you from the associated service, but does not necessarily cause the processes you were using on that service to be terminated. For this reason, it is usually better to log off of the service instead of using DELC. If the service does not disconnect you from CDCNET after you log off, use the DELC command.

- This command can be used to delete a connection when you have reached the user connection limit and need to create another connection. This command can also be used to discontinue a logon sequence in the event that you create a connection to a service that you cannot, or don't care to, log onto.

- This command can be used to delete a passthrough connection (refer to chapter 7).

- If you delete the only remaining connection, a switched line times out (as determined by a network definition parameter) and disconnects.

- $NET can only be deleted if you are at a terminal connected by the Asynchronous and X.25 protocols. In the case of the asynchronous protocol, all connections are deleted and the line disconnected (modem signals are dropped). If you are connected to CDCNET through a front-end communication device, you may, depending on the device, still be controlled by that device. In the case of X.25 protocol, your virtual circuit is deleted and you are back under the control of the X.25 PAD (such as TELENET or TYMNET) to which you were first connected.

Examples

The following command deletes a connection named EDIT, without affecting the working connection:

```
%DELC EDIT
```

The following message is displayed to the session in progress on the working connection:

```
Connection EDIT deleted.
```

The following command deletes the working connection and connects the terminal to the default connection, $NET:

```
%DELC
```

If you do not have other service connections on hold, the working connection is deleted and CDCNET issues the following message:

```
You may enter CDCNET commands.
```

If you *do* have other service connections on hold, the working connection is deleted and CDCNET issues a message that identifies the remaining connections. This message is formatted as follows:

```
Connection_Name                Service_Name

 $A                              ABC
 EDIT                            XYZ

You may enter CDCNET commands.
```

The following command, entered on the $NET connection, disconnects your terminal from CDCNET:

```
%DELC $NET
```

If you specify an invalid CONNECTION_NAME parameter value, the network issues the following message:

```
Connection <connection_name> is unknown.
```

If you enter a DELC command on the $NET connection, but do not specify a CONNECTION_NAME parameter, the network issues the following message:

```
Parameter CONNECTION_NAME is required when
DELC is entered from the $NET connection.
```

DISPLAY_COMMAND_INFORMATION (DISCI)

Purpose

Displays the parameters for a CDCNET terminal user command. The information includes the names and abbreviations of parameters, their types (including allowed keyword values), and their default values (if the parameter is not required). This command is similar to the NOS/VE command of the same name.

Format

%DISPLAY_COMMAND_INFORMATION
COMMAND=name

Parameters

COMMAND (C)

This parameter specifies the full or abbreviated name of the command for which information is to be displayed.

Remarks

Command parameter displays from this command are written for users of the Asynchronous protocol. Not all parameters and parameter value ranges are necessarily valid for other protocols.

Examples

The following command displays information about the CREATE_CONNECTION command:

```
% DISCI CREC
```

Output from this command is formatted as follows:

```
service_name, sn      : name = $required
connection_name, cn : name = $optional
output_action, oa     : key discard, d, hold, h = hold
wait, w               : boolean = false
service_data, sd      : string, 1...63 = $optional
```

Each line of the display provides information about one of the command's parameters. The parameter name, and its abbreviation, appear to the left of the colon (:). Valid parameter values are listed to the right of the colon. Default values are listed to the right of the equal sign (=).

For example, the third parameter of the CREC command (see above) is OUTPUT_ACTION, or OA. Its value is of type *key*, with the allowed values *discard* (or *d*), and *hold* (or *h*). The default value for the OA parameter is *hold*.

DISPLAY_COMMAND_LIST (DISCL)

Purpose Displays an alphabetically ordered list of the valid CDCNET terminal user commands, which includes all of the commands documented in this chapter.

Format **%DISPLAY_COMMAND_LIST** or **%HELP**

Examples The following command displays a list of the CDCNET commands that can be entered from a terminal:

```
%DISCL
```

Output from this command is formatted as follows:

```
activate_auto_recognition
activate_x_personal_computer
change_connection_attribute
change_terminal_attribute
change_working_connection
create_connection
delete_connection
display_command_information
display_command_list
display_connection
display_connection_attribute
display_service
display_terminal_attribute
do
help
request_network_operator
```

DISPLAY_CONNECTION_ATTRIBUTES (DISCA)

Purpose

Displays your connection attributes. If you do not list one or more of the attributes, the network displays the complete list.

Format

%DISPLAY_CONNECTION_ATTRIBUTES
DISPLAY_OPTION=connection attribute or
CONNECTION_ATTRIBUTE=connection attribute

Parameters

DISPLAY_OPTION (DO) or *CONNECTION_ATTRIBUTE (CA)*

Specifies a connection attribute. See the connection attribute descriptions in chapter 4.

Remarks

- The network displays connection attributes in the order that you list them. If you do not specify any attributes, all are listed alphabetically.
- You can use the DISPLAY_CONNECTION_ATTRIBUTES command on any connection except $NET.

Examples

Entering a single connection attribute with the DISPLAY_CONNECTION_ATTRIBUTES command produces the following type of display:

```
%DISCA ACA

Attention_Character_Action     : 2
```

You can also specify a list of attributes as shown in the following example:

```
%DISCA (ACA BKA IBS)

Attention_Character_Action     : 2
Break_Key_Action               : 0
Input_Block_Size               : 160
```

DISPLAY_CONNECTIONS (DISC)

Purpose

Displays your terminal's existing CDCNET connections and identifies the current working connection. The display also names your terminal and the device interface (DI) system through which your CDCNET connections are managed. This information can be very useful to your site administrator or network operator when you contact them about a connection problem.

Format

%DISPLAY_CONNECTIONS

Remarks

- The DISPLAY_CONNECTIONS command identifies your connections and equipment. You can use this command to recall connection names, to review how the network handles output from each connection, and to check your connection limit.
- The display presents information about your working connection. Next, it lists all existing connections, including your working connection.

  ($NET may be the *working* connection, but is not listed as an *existing* one.)
- The CONNECTION_STATUS line in the output display is useful to monitor a connection that you made to an unavailable service, with the WAIT parameter set to YES.

Examples

The following command displays information about the current status of connections to CDCNET:

```
%DISC
```

Output from this command is formatted as follows:

```
DI System Name         : 080025100081, TDI_81
Terminal Name          : 5300000000,$CONSOLE_100081_5300000000
Working_Connection     : EDIT
User_Connection_Limit  : 4
Status_Action          : SEND
Connection_Name        : $A
Connection_Status      : PENDING(01)
Service_Name           : ABC
Output_Action          : HOLD(0)

Connection_Name        : EDIT
Connection_Status      : CONNECTED
Service_Name           : XYZ
Output_Action          : SEND
```

The first part of this display is composed of five lines that identify your working connection and supply additional information about network communications.

DI System Name

Identifies the device interface (DI) you are using. This line of the display has two parts: system identifier (080025100081) and its logical system name (TDI_81). The first is a unique network name. The second is an optional, site-specified name.

Both names identify the same components. They also appear in the CDCNET banner when you sign on. You should make a note of them in case you need to report communications problems.

Terminal Name

Displays the name that the network uses for your terminal (5300000000,$CONSOLE_100081_5300000000). Like the DI system name, it is made up of two parts: the network name (5300000000) and the logical name ($CONSOLE_100081_5300000000).

The network name is unique, and indicates how your terminal is connected to the network: LIM number (5), port (3), cluster address (0000), and device address (0000).

The site administrator can specify a logical name when your terminal device is configured on the network; otherwise, the network assigns a default logical name. A default logical name is composed of the following three parts:

- The first part ($CONSOLE) identifies the type of terminal used.
- The second part (100081) is made up of the last six digits of the DI system identifier.
- The third part (5300000000) is the same as your terminal's network name.

Working Connection

Identifies the connection you are currently using (EDIT, in the example). This connection (unless it is $NET) is also listed in the second part of the display as one of your existing connections.

User Connection Limit

Displays the number of simultaneous connections (4) the network administrator permits you to have in addition to $NET.

Status Action

Reports the setting of the STATUS_ACTION terminal attribute for the working connection (SEND). Possible entries are DISCARD, HOLD, and SEND.

The second and subsequent parts of the display list your existing connections. Each part includes the connection, its service, and the OUTPUT_ACTION for that connection. In the example, you have two connections ($A and EDIT).

Connection Name

Identifies the name either you or the network assigned this connection when it was created.

Connection Status

Reports the status of connections created. Possible values are CONNECTED, DISCONNECTED, and PENDING. The DISCONNECTED status identifies connections that have timed out or have disconnected but have output being held. PENDING is displayed when a connection to an unavailable service is made with WAIT parameter specified to YES.

Next, the network shows, in parentheses, how many users have been waiting for the service to become available. A value of 01 means you are the next user to get connected to the service when it becomes available. A value of 99 means 99 or more users are waiting for the service to become available.

Service Name

Identifies the service at the other end of the connection.

Output Action

Reports the last setting of STATUS_ACTION that you specified on a CREATE_CONNECTION or CHANGE_WORKING_CONNECTION command for each connection, except your working connection. The network uses SEND for the output action on the working connection. Possible entries for the other connections are DISCARD and HOLD. If the output action is HOLD, the network also gives the number of messages held, in parentheses.

DISPLAY_SERVICE (DISS)

Purpose Displays information about the set of services selected for display by the site administrator, including service availability. These are some of the service names that you can specify on a CREATE_CONNECTION command. The display shows service names, service status, and other information.

Format **%DISPLAY_SERVICE or DISPLAY_SERVICES**
SERVICE=service name or *keyword*
DISPLAY_OPTION=display option

Parameters *SERVICE (S)*

Identifies the service(s) about which you want information. If you specify more than one service for display, enclose the names in parentheses and separate them with spaces. If you omit this parameter, the network displays information about all of the services selected by the site administrator for display.

DISPLAY_OPTION (DO)

Specifies one of the following entries:

BRIEF	Displays status for each service name listed.
FULL	Displays status and additional information for each service name listed.

Default is FULL.

Remarks

The network assigns each service one of the following status options:

Option	Description
UP	Service is available.
DOWN	Service is not available.
BUSY	Service is available, but needed resources are currently in use. Try again later.
UNKNOWN	The network does not recognize the service name you specified. Either the site administrator has not selected the name for display, or you entered an incorrect name.

NOTE

Whenever a CREATE_CONNECTION command is entered for a given service (not including connections made using an alias), the network updates data used in creating the DISPLAY_SERVICE display for that service. The network also updates this data at intervals specified by your site administrator. This command provides accurate information only to the extent that service status changes have been updated in one of these two ways.

Examples

The following command displays information in the brief format about services VE_001 and VE_002:

```
%DISS S=(VE_001 VE_002) DO=BRIEF

VE_001                                    :  UP
VE_002                                    :  UP
```

If you enter DISS without parameters, the network alphabetically lists all services that your site administrator specified for the display. The display is in full format, so it includes any special information your site administrator specified for the display, such as alternate names or troubleshooting messages. For example:

```
VE_001                                    :  UP
 Alternate title   : RED.
VE_002                                    :  UP
 Alternate title   : GREEN.
VE_003                                    :DOWN
 Alternate titles : BLUE, NAOMI, LAB.
```

For each service, the first line displayed is the same as what would be displayed in the brief format. Any additional lines of display for a service are specified for display by the site administrator.

DISPLAY_TERMINAL_ATTRIBUTE (DISTA)

Purpose Displays your terminal's characteristics. The complete list is displayed if you do not list one or more attributes.

Format **%DISPLAY_TERMINAL_ATTRIBUTE**
DISPLAY_OPTION=terminal attribute or
TERMINAL_ATTRIBUTE=terminal attribute

Parameters *DISPLAY_OPTION (DO)* or *TERMINAL_ATTRIBUTE (TA)*

Specifies a terminal attribute. See the terminal attribute descriptions in chapter 3.

Remarks The network displays your attributes in the order that you list names in the DISPLAY_OPTIONS parameter. If you do not specify any names, all attributes except CONTROL_CODE_REPLACEMENT (CCR) are listed alphabetically. CCR is displayed only when the command is entered with DISPLAY_OPTION set to ALL or CCR.

Examples The following example shows how you can display a selected list of attributes using the DISPLAY_TERMINAL_ATTRIBUTE command.

```
%DISTA (CRS CS LFS)

Carriage_Return_Sequence       :CR
Code_Set                       :ASCII
Line_Feed_Sequence             :LF
```

If you prefer, you can display one attribute at a time.

```
%DISTA CRS

Carriage_Return_Sequence       :CR
```

DO

Purpose

Executes a network procedure. (Refer to the CDCNET Configuration Guide for more information about network procedures.)

Format

%DO
PROCEDURE_NAME=procedure name
PROCEDURE_TYPE=procedure type

Parameters

PROCEDURE_NAME (PN)

Specifies the procedure name.

PROCEDURE_TYPE (PT)

Identifies the type of procedure.

TDP	Specifies a terminal definition procedure.
TUP	Specifies a terminal user procedure.

Default is TUP.

Remarks

The site administrator might configure your network so that it automatically executes any procedure needed to access a service.

Examples

The following command executes a terminal user procedure called SERVIC1:

```
%DO SERVIC1
```

REQUEST_NETWORK_OPERATOR (REQNO)

Purpose Sends a message from your terminal to the network operator.

Format **%REQUEST_NETWORK_OPERATOR**
MESSAGE = string

Parameters **MESSAGE (M)**

Specifies a 1- through 15-line message. Each line is enclosed in single quotes. If more than one line is to be sent, enclose the lines in parentheses and separate lines from each other with a space. You can use as many as 255 characters in the entire command.

Remarks

- When you come to the end of the physical line on your screen while entering the MESSAGE parameter, your terminal (or CDCNET) folds the line so that you can continue typing the message on the next line.
- CDCNET tells you when it sends your message. However, it does not tell you if it successfully delivered the message, or whether a network operator received the message. The network logs the message even if no network operator is connected to CDCNET when your message arrives.

Examples The following command sends three lines of a message to the network operator:

```
%REQNO M=('What is the new holiday schedule?' 'Will
the laser printer be available then?' 'When will the
overnight printing be completed?')
```

The network operator sees the following message:

```
What is the new holiday schedule?
Will the laser printer be available then?
When will the overnight printing be completed?
```

Terminal Attributes 3

Terminal Attributes 3

Input and output from your terminal are handled by CDCNET according to a set of characteristics called terminal attributes. Every terminal connected to CDCNET has one set of terminal attributes. These attributes maintain a constant value across all connections.

Displaying and Changing Terminal Attributes

You can display and change your terminal attributes with the following two CDCNET commands:

DISPLAY_TERMINAL_ATTRIBUTE

CHANGE_TERMINAL_ATTRIBUTE

The DISPLAY_TERMINAL_ATTRIBUTE command lists all your terminal's attribute settings or just selected ones. Use this command to review the list or verify that a particular setting is correct.

Normally, you can access CDCNET, create a service connection, and begin using that service with the default terminal attribute settings. If your terminal or your processing require that you use different settings, use the CHANGE_TERMINAL_ATTRIBUTE command to make the changes.

The DISPLAY_TERMINAL_ATTRIBUTE command lists only those attributes supported by your terminal's protocol; the CHANGE_TERMINAL_ATTRIBUTE command can be used to change any terminal attribute, even if the attribute is not supported by the protocol. A nonsupported attribute cannot be displayed or validated by your terminal's protocol and has no effect on your terminal. .

Both of these terminal attribute commands are described in chapter 2.

Terminal Attributes and Protocols

Figure 3-1 shows the default terminal attribute settings for the Asynchronous protocol. For the default terminal attribute settings for the 3270 Binary Synchronous Communications, X.PC, Mode 4, and 3270 SNA communications protocols see appendixes F, H, I, and J, respectively.

```
Attention_Character          :NUL
Backspace_Character          :BS
Backspace_Window             :PW
Begin_Line_Character         :NUL
Cancel_Line_Character        :CAN
Carriage_Return_Delay        :0
Carriage_Return_Sequence     :CR
Character_Flow_Control       :ON
Code_Set                     :ASCII
Echoplex                     :OFF
End_Line_Character           :CR
End_Line_Positioning         :LFS
End_Output_Sequence          :
End_Page_Action              :NONE
End_Partial_Character        :LF
End_Partial_Positioning      :CRS
Fold_Line                    :ON
Form_Feed_Delay              :0
Form_Feed_Sequence           :FF
Function_Key_Class           :NONE
Hold_Page                    :OFF
Hold_Page_Over               :ON
Line_Feed_Delay              :0
Line_Feed_Sequence           :LF
Network_Command_Character    :%
Page_Length                  :24
Page_Width                   :80
Parity                       :EVEN
Response_Action              :SEND
Status_Action                :SEND
Terminal_Model               :NONE
```

Figure 3-1. Asynchronous Protocol Terminal Attribute Defaults

Terminal Attributes and Input Editing Modes

The terminal attributes that apply to your input and output processing depend upon whether the INPUT_EDITING_MODE connection attribute is set to NORMAL or TRANSPARENT. See table 3-1. Chapter 5 describes normal and transparent modes in more detail.

Table 3-1. Terminal Attributes and Input Editing Modes

Terminal Attribute	Normal	Transparent
Attention_Character	I	I
Backspace_Character	I	
Backspace_Window	I	
Begin_Line_Character	I	
Cancel_Line_Character	I	
Carriage_Return_Delay	I/O	
Carriage_Return_Sequence	I/O	I
Character_Flow_Control	I/O	I/O
Code_Set	I/O	
Control_Code_Replacement	O	O
Echoplex	I	I
End_Line_Character	I	
End_Line_Positioning	I	
End_Output_Sequence	O	O
End_Page_Action	O	
End_Partial_Character	I	
End_Partial_Positioning	I	
Fold_Line	O	
Form_Feed_Delay	O	
Form_Feed_Sequence	O	
Function_Key_Class		I
Hold_Page	O	
Hold_Page_Over	O	
Line_Feed_Delay	I/O	
Line_Feed_Sequence	I/O	I
Network_Command_Character	I	
Page_Length	O	

(Continued)

Table 3-1. Terminal Attributes and Input Editing Modes *(Continued)*

Terminal Attribute	Normal	Transparent
Page_Width	I/O	
Parity	I/O	I/O
Response_Action	O	
Status_Action	O	
Terminal_Model[1]		

Attributes marked I apply to input. Those marked O apply to output.

1. This attribute does not affect input or output processing.

Terminal Attribute Descriptions

The following pages describe the terminal attributes provided by CDCNET for use with the Asynchronous protocol. Refer to appendixes F through I for terminal attribute support for other protocols.

The individual attribute descriptions contain purpose, format, remarks, and examples, when applicable. Each description begins with an attribute name followed by its abbreviation enclosed in parentheses. The format description lists the values that you can assign a specified terminal attribute. It also provides an explanation of each value and its abbreviation.

ATTENTION_CHARACTER (AC)

Purpose Identifies the key you press to perform the action specified by ATTENTION_CHARACTER_ACTION. (See chapter 4 for a description of the ATTENTION_CHARACTER_ACTION connection attribute and chapter 6 for more information on interrupts.)

Format *ATTENTION_CHARACTER=character or NUL*

character — Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

NUL — Specifies that no character is recognized as an attention character.

The default is NUL.

Remarks

- The ATTENTION_CHARACTER can serve as a substitute for the BREAK key when your terminal does not have a BREAK key.
- The specified attention character action is executed whenever the character is received from your terminal; the character does not have to occur at the beginning of a line.
- The terminal attribute ATTENTION_CHARACTER is interpreted as data when the connection attribute ATTENTION_CHARACTER_ENABLE=NO.
- You can perform different actions with the BREAK key and ATTENTION_CHARACTER if you already have a BREAK key. You define these actions by specifying different values for the ATTENTION_CHARACTER_ACTION and BREAK_KEY_ACTION connection attributes.
- CDCNET recognizes the attention character irrespective of the parity bit setting and the PARITY terminal attribute setting.
- When CHARACTER_FLOW_CONTROL (described later in this chapter) is ON, CDCNET treats the ATTENTION_CHARACTER is an implied X-ON.

Examples

Suppose you want a quick way to execute a level 2 user interrupt. You can create one by setting your ATTENTION_CHARACTER_ACTION connection attribute to 2, and then setting your ATTENTION_CHARACTER terminal attribute. For example, the following command sets your attention character to ASCII DC4, which can be entered by holding down the CONTROL key and pressing T.

```
%CHATA AC=20
```

After this command executes, CDCNET recognizes received values of 14(16) and 94(16) as the attention character.

BACKSPACE_CHARACTER (BC)

Purpose

Identifies the key you press to backspace over (delete) the last character(s) you entered on your current line.

Format

BACKSPACE_CHARACTER=character

character — Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

The default is BS.

Remarks

- You can only delete characters with a BACKSPACE key when these connection attribute settings apply to your connection.

  INPUT_EDITING_MODE=NORMAL
  STORE_BACKSPACE_CHARACTER=OFF

  With these settings, the character you entered by mistake and the BACKSPACE_CHARACTER are not sent to your service as input.

- If the backspace character you specify for CDCNET does not match the terminal's backspace character, you cannot overstrike on the screen or page: what you see is not what your service receives.

  If the terminal's backspace character and BACKSPACE_CHARACTER are the same, the service receives what you see.

- You can backspace until you encounter a backspacing boundary, from which no further backspacing is allowed. This boundary is either the beginning of the current input, or the character position in the current input equal to the BACKSPACE_WINDOW, whichever comes first. The boundary moves as each BACKSPACE_WINDOW number of characters are input.

- You cannot backspace over characters that forward data: END_PARTIAL_CHARACTER (when PARTIAL_CHARACTER_FORWARDING is set to ON) and END_LINE_CHARACTER.

Examples

If you enter the following characters and then realize you have typed Q instead of C, you can correct the mistake.

```
ABQ
```

Press the BACKSPACE key (often a reverse arrow) and enter C. As with all entries, press the RETURN key (or equivalent) to send your entry to the service. If your terminal's backspace character is the same as the BACKSPACE_CHARACTER, you see the following on your screen.

```
ABC
```

As a result of your entry, CDCNET sends the following characters to your service.

```
ABC
```

Now change your BACKSPACE_CHARACTER (CDCNET) to an exclamation point (!) so that it is different from your terminal's backspace character.

```
%CHATA BC='!'
```

Reenter the first example using your newly defined BACKSPACE key. To do this, retype the original characters, ABQ, backspace using the exclamation point, and then enter a C. The following series of characters, including the new BACKSPACE_CHARACTER, appears on your screen.

```
ABQ!C
```

The difference between what is shown on your screen in these examples, the BACKSPACE_CHARACTER, does not affect the actual characters your service receives.

```
ABC
```

In both examples, CDCNET edits your input and removes the BACKSPACE_CHARACTER and the preceding character before it forwards the input.

A third example illustrates how far you can backspace. Suppose that your terminal has a PAGE_WIDTH of 80 characters and a BACKSPACE_WINDOW of 120 characters. You can enter up to 120 characters of input and backspace over any of them. Once 120 characters are input, you cannot backspace past the 120th character. In this example, by setting BACKSPACE_WINDOW greater than PAGE_WIDTH, input can be backspaced over past a PAGE_WIDTH boundary.

A fourth example is with a PAGE_WIDTH of 80 characters and a BACKSPACE_WINDOW of 30 characters. You can enter up to 30 characters and backspace over any of them. Once 30 characters are input, you cannot backspace past the 30th character. You can continue to input up to 30 more characters and backspace over them. In this example, the portion of your input over which you can backspace changes as each BACKSPACE_WINDOW number of characters are input.

BACKSPACE_WINDOW (BW)

Purpose

Defines the maximum number of characters on the current input line that can be backspaced over using the BACKSPACE_CHARACTER (BC). That is, you can backspace across an input line until you encounter a backspacing boundary, which is either the beginning of the current input line, or the character position specified as the BACKSPACE_WINDOW, whichever comes first.

Format

BACKSPACE_WINDOW=integer

integer	Specifies the character position of the backspace boundary. You can use any integer from 10 through 255. If you enter 0, the network uses the value of the PAGE_WIDTH terminal attribute (if the PAGE_WIDTH attribute is 0, the network uses 255).

The default is PW.

Remarks

If CDCNET is echoing your entries and you try to backspace past the BACKSPACE_WINDOW, CDCNET does not move your cursor back any further, because you have already "edited" the specified maximum number of characters.

If your terminal is echoing the entries, you might see the cursor move past the BACKSPACE_WINDOW boundary, but CDCNET ignores the additional backspacing.

Examples

See the preceding BACKSPACE_CHARACTER description for examples of using BACKSPACE_WINDOW.

BEGIN_LINE_CHARACTER (BLC)

Purpose Identifies a character that your terminal automatically transmits at the beginning of every input line. When CDCNET receives this character, it discards the character.

Format *BEGIN_LINE_CHARACTER=character*

character — Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

The default is NUL.

Remarks

- Some terminals automatically send a special character after each END_LINE_CHARACTER and before any additional input.
- If you do not want your service to receive this special character, you can have CDCNET discard it before forwarding input data to your service. Specifying the character as the BEGIN_LINE_CHARACTER ensures that CDCNET discards it.

Examples If you are working at a terminal that allows you to edit a screen of information locally before sending it to CDCNET, you can change BEGIN_LINE_CHARACTER to STX.

```
%CHATA BLC=2
```

Defining BEGIN_LINE_CHARACTER, as a special character that precedes each screen, enables you to edit an entire screen before forwarding any information. Since CDCNET removes STX from your input, it does not affect communications between CDCNET and your service.

CANCEL_LINE_CHARACTER (CLC)

Purpose Identifies the key you press to cancel the line you are currently typing.

Format *CANCEL_LINE_CHARACTER=character or NUL*

character — Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

NUL — Specifies that no character cancels a line.

The default is CAN.

Remarks

- Cancelling a line involves only the line you are currently entering. It does not affect output being displayed or printed at your terminal.
- In order to correct a mistake you have just made, press these two keys: the CANCEL_LINE_CHARACTER, followed by the END_LINE_CHARACTER. CDCNET then forwards both the line you are removing and a cancel indication. Your service defines the action taken by the indication.

  CDCNET also sends your terminal the following message.

  ```
  Input cancelled.
  ```

Examples When you use the following attribute settings, you may want to change your CANCEL_LINE_CHARACTER so that your terminal echoes the CANCEL_LINE_CHARACTER on the screen.

CANCEL_LINE_CHARACTER=CAN
END_LINE_CHARACTER=CR

To do this, change the CANCEL_LINE_CHARACTER to an ampersand (&).

```
%CHATA CLC='&'
```

Then enter some data to test your new CANCEL_LINE_CHARACTER.

```
ABCDE
```

Cancel the incomplete line by entering an ampersand and then press the RETURN key.

```
ABCDE&
```

The network forwards the line with a cancel indication to your service. It also sends you this message:

```
Input cancelled.
```

However, if you mistype and enter an ampersand followed by a *space* and then press the RETURN key, the line is forwarded without the cancel indication. When the CANCEL_LINE_CHARACTER is followed by a space and not an END_LINE_CHARACTER, CDCNET treats the entire line, including the ampersand, as (uncancelled) data. As a result, the network forwards it to your service.

CARRIAGE_RETURN_DELAY (CRD)

Purpose

Increases or decreases idle time after a CARRIAGE_RETURN_SEQUENCE. (During this time, the network sends your terminal NUL characters.)

If information is sent to your terminal faster than it can position the carriage (or cursor), you can avoid losing data by increasing the idle time with CARRIAGE_RETURN_DELAY. This additional time allows your terminal and any slave printer to perform a return before more output data is transmitted. (See chapter 5 and the description of the CARRIAGE_RETURN_SEQUENCE terminal attribute later in this chapter for more information.)

NOTE

Carriage return delays are typically used for older, nonbuffered devices that do not support X-ON/X-OFF flow control.

Format

CARRIAGE_RETURN_DELAY=integer

integer	Specifies the idle time in number of milliseconds. You can use any integer from 0 through 1000.

The default is 0.

Remarks

- Whenever any of the following occurs, CDCNET inserts enough NUL characters in the output to fill the milliseconds you specify with CARRIAGE_RETURN_DELAY:
  - Output from your service contains a character (such as a format effector or CR) that the network converts to the CARRIAGE_RETURN_SEQUENCE.
  - PAGE_WIDTH is reached; the NUL characters for the delay are sent whether it is CDCNET or your terminal that folds the line.

- The network performs END_LINE_POSITIONING or END_PARTIAL_POSITIONING and, as a result, sends your terminal a CARRIAGE_RETURN_SEQUENCE.

- A combination of factors determines the delay you enter for the CARRIAGE_RETURN_DELAY. The amount of time needed to reposition the carriage (or cursor) depends upon the terminal's page width, the speed at which the device moves the carriage, and line speed.

Factor	Delay Time
Page width increases.	Increases.
Device performs cursor positioning faster.	Decreases.
Line speed increases.	Decreases.

- Certain terminals, such as the Lear Siegler ADM3A and ADM5, do not ignore NUL characters. In these cases, you should set CARRIAGE_RETURN_DELAY to 0.

- Using both the CARRIAGE_RETURN_DELAY and LINE_FEED_DELAY provides a maximum combined delay of 2 seconds for repositioning, if needed.

Examples

Slave printers that repeat information exchanged by CDCNET and a terminal are often attached to microcomputers. In this situation, you can avoid losing data on your printed listing by changing the CARRIAGE_RETURN_DELAY.

```
%CHATA CRD=200
```

Specifying a delay of one-fifth second provides additional time for repositioning the carriage. During the time CDCNET is inserting NUL characters after each CARRIAGE_RETURN_SEQUENCE, your printer is repositioning its carriage.

CARRIAGE_RETURN_SEQUENCE (CRS)

Purpose

Defines the sequence of characters that positions the cursor or carriage at the beginning of a line. (See the descriptions of the END_LINE_POSITIONING and END_PARTIAL_POSITIONING terminal attributes later in this chapter for more information. Also see chapter 5.)

Format

CARRIAGE_RETURN_SEQUENCE=sequence

sequence — Defines a sequence of 0 through 2 characters. (See appendix B for detailed information about defining characters.)

The default is CR.

Remarks

- CDCNET sends a CARRIAGE_RETURN_SEQUENCE to your terminal whenever a CR (ASCII character) appears in your data.
- When the service uses a format effector that requires a return, CDCNET substitutes the sequence defined by CARRIAGE_RETURN_SEQUENCE.
- CDCNET inserts a CARRIAGE_RETURN_SEQUENCE when PAGE_WIDTH is reached and either CDCNET or your terminal folds the line (depending upon the setting of FOLD_LINE).

Examples

If your terminal automatically performs both a line feed and carriage return, you can set the CARRIAGE_RETURN_SEQUENCE to NUL to avoid duplication. To do so, enter the following CHANGE_TERMINAL_ATTRIBUTE command.

```
%CHATA CRS=''
```

CHARACTER_FLOW_CONTROL (CFC)

Purpose Specifies whether your terminal and CDCNET control the flow of data using X-ON/X-OFF protocol (DC1 and DC3 characters).

You can lose data if CHARACTER_FLOW_CONTROL is set to OFF and if your terminal cannot operate at high speeds. (See chapter 5 for more information.)

Format

CHARACTER_FLOW_CONTROL=ON
CHARACTER_FLOW_CONTROL=OFF
CHARACTER_FLOW_CONTROL=INPUT
CHARACTER_FLOW_CONTROL=OUTPUT

ON — Uses the X-ON/X-OFF protocol to regulate both input data from your terminal and output data to your terminal.

OFF — Does not use the X-ON/X-OFF protocol to regulate the flow of data to/from your terminal.

INPUT — Uses the X-ON/X-OFF protocol to regulate the flow of input data from your terminal.

OUTPUT — Uses the X-ON/X-OFF protocol to regulate the flow of output data to your terminal.

The default is ON.

Remarks

- The X-ON/X-OFF protocol controls movement of both input and output data with ASCII characters.

Protocol	ASCII Mnemonic	Control Character	Meaning
X-ON	DC1	Q	Send more data.
X-OFF	DC3	S	Stop sending data.

- To use CHARACTER_FLOW_CONTROL effectively, you must know how your terminal handles flow control.

  If your terminal supports automatic flow control, it may use the X-ON/X-OFF protocol, EIA flow control, or both. Most terminals support the X-ON/X-OFF protocol.

- If your site administrator has defined the communication line you are using with EIA flow control turned on, CDCNET uses EIA flow control to control your input regardless of the CFC setting.

- When CHARACTER_FLOW_CONTROL is ON, CDCNET does the following:

  - Recognizes DC1 and DC3 characters as flow control from your terminal, starts and stops sending output data accordingly, and discards the DC1 and DC3 characters.

  - Sends DC1 and DC3 characters to your terminal to tell it to start and stop sending input data.

- When CHARACTER_FLOW_CONTROL is OFF, CDCNET does the following:

  - Forwards any DC1 and DC3 characters that it receives as input data to the service.

  - Does not send DC1 and DC3 characters to your terminal for flow control.

- When CHARACTER_FLOW_CONTROL is ON and the network has sent the terminal a DC3 character to stop the flow of input, the network is unable to handle more characters. The network continues to accept a line-speed dependent number of input characters after sending the DC3 character to the terminal. When this number is reached, each subsequent character that the terminal sends the network is discarded and a BEL character is sent to the terminal instead.

- When CFC=ON, CDCNET recognizes received values 11(16) and 91(16) as X-ON signals, and 13(16) and 93(16) as X-OFF signals. Therefore, CDCNET recognizes X-ON and X-OFF characters regardless of the parity bit setting and the PARITY terminal attribute setting.

- When CFC=ON, CDCNET transmits X-ON and X-OFF characters with the parity bit set as specified by the PARITY terminal attribute setting.

- If the service sends DC1 or DC3 characters to your terminal, CDCNET does not detect them and does not edit them out. The effect of the service sending DC1 or DC3 characters is terminal dependent.

- When CHARACTER_FLOW_CONTROL is ON, the DC1 and DC3 characters that the network generates to regulate output are based on the PARITY setting. If this setting does not match the terminal's setting, some terminals such as the CDC 721 ignore DC1 and DC3 characters.

  If this happens, it may appear that the network has not changed the CHARACTER_FLOW_CONTROL setting. Instead, CHARACTER_FLOW_CONTROL seems to remain either ON or OFF. This problem is most likely to occur when you are using automatic recognition, and the terminal's setting is not even or odd. If it is none, change PARITY to NONE. If the terminal's setting is different, experiment with various PARITY settings to determine what works best for your terminal.

- When CHARACTER_FLOW_CONTROL is ON, the ATTENTION_CHARACTER and BREAK key function implicitly as X-ON.

Examples

When you use a file transfer protocol, such as XMODEM, it disables flow control during a file transfer. The protocol accomplishes this internally with a terminal redefinition command. You can accomplish the same by changing CHARACTER_FLOW_CONTROL to OFF via the following command.

```
%CHATA CFC=OFF
```

Some file transfer protocols do not use X-ON/X-OFF, permitting you to transfer DC1 and DC3 as data within a file.

CODE_SET (CS)

Purpose

Identifies the coded character set that your terminal uses (usually ASCII).

If you receive garbled (unreadable) messages when you try to access the network, the communication line that connects your terminal to the computer site probably requires a different coded character set (or parity setting). At this point, you have two choices:

1. Call your site administrator to determine your terminal's proper setting. Then enter a CHANGE_TERMINAL_ATTRIBUTE command with your terminal's setting for CODE_SET (or PARITY).

2. Turn off your terminal and consult your terminal documentation to verify the setting.

If necessary, reset your terminal switch to match the characteristics of your communication line.

Format

CODE_SET=ASCII
CODE_SET=ASCII48
CODE_SET=ASCII64
CODE_SET=ASCII95
CODE_SET=ASCII128
CODE_SET=ASCII256
CODE_SET=BPAPL
CODE_SET=EBCDIC
CODE_SET=TPAPL
CODE_SET=name

ASCII	Uses the 7-bit ASCII-coded character set. ASCII-coded character set is identical to the ASCII128-coded character set.
ASCII48	Uses a translation table to map the 7-bit ASCII-coded character set to ASCII48. See CDCNET Configuration Guide for the translation table.
ASCII64	Uses a translation table to map the 7-bit ASCII-coded character set to ASCII64. See CDCNET Configuration Guide for the translation table.

ASCII95	Uses a translation table to map the 7-bit ASCII-coded character set to ASCII95. See CDCNET Configuration Guide for the translation table.
ASCII128	Uses the 7-bit ASCII-coded character set. This is identical to the ASCII-coded character set.
ASCII256	Uses the 8-bit ASCII-coded character set.
BPAPL	Uses the bit-pairing ASCII-coded character set with APL print.
EBCDIC	Uses the 8-bit EBCDIC-coded character set.
TPAPL	Uses the typewriter-pairing ASCII-coded character set with APL print.
name	Uses the name of a site-defined code set.

The default is ASCII.

Remarks

- The following table shows how the CODE_SET attribute affects your terminal's input and/or output. This table gives information about only interactive input and output. For information about batch input and output, see the CDCNET Site Configuration Guide.

Input and Output	Output Only
ASCII 128	ASCII 48[1]
ASCII 256	ASCII 64[1]
BAPL	ASCII 95[1]
EBCDIC	Site-defined code sets[2]
TPAPL	Site-defined code sets[2]

- Most terminals operate with ASCII; however, if you need to work with APL and if your terminal handles APL, you can switch to an APL character set. Depending on how your terminal attributes are set and which ones are relevant to your APL work, you can use the ACTIVATE_AUTO_RECOGNITION or CHANGE_TERMINAL_ATTRIBUTE command.

- Your site may have defined its own code set. Call your site administrator for the code set name. See CDCNET Site Configuration Guide for information on how to define a code set for your site.

1. The associated input code-set for these is ASCII 128.

2. The associated input code-set for these depends on the terminal's protocol. For example, the associated input code-set for a 3270 terminal is EBCDIC.

Examples

Suppose you have a Bubble APL terminal that you normally use with the coded character set switch on ASCII. To use the NOS APL interpreter, you need to change the NETWORK_COMMAND_CHARACTER and ATTENTION_CHARACTER as well as this setting and CODE_SET.

1. Enter the command:

```
%CHATA NCC=ESC AC=DC4 CS=TPAPL
```

2. Set your terminal's switch on APL to select the typewriter-pairing ASCII coded character set with APL print.

When you have completed your APL work and want to use ASCII again, repeat the steps.

1. Enter the command:

```
ESC CHATA NCC=% AC=NUL CS=ASCII
```

2. Set your terminal's switch on ASCII.

In the following example, a site-defined code set named ALPHALOW is being specified as a value for the CODE_SET parameter. ALPHALOW translates all uppercase ASCII alpha characters to lowercase alpha characters.

```
%CHATA CS=ALPHALOW
```

CONTROL_CODE_REPLACEMENT (CCR)

Purpose Specifies a list of control and replacement code pairs.

Format *CONTROL_CODE_REPLACEMENT=list*

list Defines a list of 1 through 64 control code and/or replacement code pairs.

Remarks

- The first element of each pair contains a range of one or more control codes that are not to be transmitted to the terminal. The second element of the pair is optional and specifies the replacement code which is to be transmitted to the terminal instead of the specified control code(s). When the second element is omitted, the control code is discarded and no replacement code is transmitted.

  The acceptable values for the first element of this pair are code values in the ranges 0 through 31 and 128 through 159. The permitted value for the second element is any value in the range 0 through 255.

- The CCR attribute affects both transparent and nontransparent output.

- Replacement codes apply only to output received from the connected service and not to output generated by CDCNET, such as LINE_FEED_SEQUENCE characters and END_OUTPUT_SEQUENCE characters.

- Replacement code cannot be applied to the backspace (BS) control character on terminals using asynchronous protocol.

- The DISPLAY_TERMINAL_ATTRIBUTE (DISTA) command displays the value of the terminal attribute CONTROL_CODE_REPLACEMENT only when specifically selected by the DO=CCR or DO=ALL parameter values of the DISTA command. (See DISPLAY_TERMINAL_ATTRIBUTE command description in chapter 2 for more information.)

Examples

In the following command, BEL code is discarded from the output and codes SOH through ACK are replaced by SP codes.

```
%CHATA CCR=( BEL,  (SOH..ACK, SP))
```

The following command is rejected because it contains BEL as the first element of one list and the second element of another list.

```
%CHATA CCR=( BEL,  (FS,BEL))

No attributes changed.  Conflict in the use of control code
BEL.
```

ECHOPLEX (E)

Purpose Specifies whether information you enter from your keyboard is sent back to your terminal by the network.

Format *ECHOPLEX=ON*
ECHOPLEX=OFF

ON Displays input at terminal. This setting is overridden by the connection attribute setting ECHO_ENABLE=NO. (See chapter 4 for a description of the ECHO_ENABLE connection attribute.)

OFF Does not display input at terminal.

The default is OFF.

Remarks

- When you enter data at your terminal, the terminal, a modem, CDCNET, or the service echoes it back to you. That is, data is displayed or printed so that you can see your entries.
- If CDCNET or the service echoes your entries instead of your terminal or modem, you may notice a short pause between pressing a key and seeing the character appear on the screen.
- You can control whether or not your terminal echoes if it has a duplex switch. Setting the switch to HALF means the terminal echoes; FULL means it does not.

  How you set ECHOPLEX and the terminal duplex switch controls echoing. Specifying these settings affects the number of times an entry is echoed.

Terminal Setting	E=ON	E=OFF
Full-duplex	Entry is echoed once.	Entry is not echoed.
Half-duplex	Entry is echoed twice.	Entry is echoed once.

- You can use character echoing by CDCNET to verify that the network is receiving your data correctly. To do this, put your terminal in full-duplex and set ECHOPLEX to ON.

- When ECHOPLEX is ON, CDCNET echoes all the characters you enter with a few exceptions:

  - If CHARACTER_FLOW_CONTROL is also ON, CDCNET does not echo DC1 and DC3 characters.

  - If connection attribute STORE_BACKSPACE_CHARACTER is OFF, CDCNET echoes BACKSPACE_CHARACTER entries until it encounters a backspacing boundary. A backspacing boundary is defined as either the beginning of the current input line, or the number of characters specified for the BACKSPACE_WINDOW terminal attribute, whichever comes first.

  If ECHOPLEX is OFF, you can backspace your cursor across a backspacing boundary. However, the additional backspacing does not affect data past the backspacing boundary.

- If CDCNET echoes, what you see on your screen is what is sent to the service. However, this is not necessarily true when the terminal echoes. If your terminal's backspace character is different than the one defined for BACKSPACE_CHARACTER, pressing the BACKSPACE key on your terminal moves the cursor, but does not affect the data being forwarded to your service.

Examples Suppose a Zenith Z29 terminal set up for full-duplex operation does not perform character echoing, and nothing you enter at the keyboard appears on the screen.

To remedy this situation, change ECHOPLEX to ON.

```
%CHATA E=ON
```

END_LINE_CHARACTER (ELC)

Purpose

Identifies the key you press to forward data you have entered for processing. This key is labeled with various symbols (for example, ENTER, NEXT, ↵, and RETURN). In this manual, the RETURN key is used to mean any key that performs this carriage-return function.

Pressing the key transmits the character code that causes CDCNET to end and forward a complete transmission to the service. (The END_LINE_CHARACTER is not treated as data.) Your service normally acts on this input. END_LINE_CHARACTER also causes CDCNET to do END_LINE_POSITIONING. (See chapter 5 for more information.)

Format

END_LINE_CHARACTER=character

character	Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

The default is CR.

Remarks

CDCNET does END_LINE_POSITIONING when it encounters an END_LINE_CHARACTER in your input by sending your terminal a sequence. This produces appropriate cursor positioning for the next line of data.

Examples

You can define END_LINE_CHARACTER as a different key than the terminal's RETURN key. For example, if your terminal has a key labeled EOT, you might want to use it. Change the ELC character with the following command:

```
%CHATA ELC=4
```

END_LINE_POSITIONING (ELP)

Purpose

Defines the character string that the network sends to your terminal to perform cursor positioning when you have entered a complete transmission. (The network sends this string after receiving an END_LINE_CHARACTER. See chapter 5 for more information.)

Format

END_LINE_POSITIONING=CRS
END_LINE_POSITIONING=LFS
END_LINE_POSITIONING=CRSLFS
END_LINE_POSITIONING=NONE

CRS	Uses the setting you specified with CARRIAGE_RETURN_SEQUENCE.
LFS	Uses the setting you specified with LINE_FEED_SEQUENCE.
CRSLFS	Uses the settings you specified with both CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE.
NONE	Does not use any character string.

The default is LFS.

Remarks

- If your terminal automatically repositions the cursor where you want it after an END_LINE_CHARACTER, END_LINE_POSITIONING is NONE.
- If your terminal does not position the cursor where you want it after an END_LINE_CHARACTER, reset END_LINE_POSITIONING. Select the setting that describes what positioning the network should do when it receives an END_LINE_CHARACTER in your input.

Examples

Suppose you access your service by going through CDCNET and a public data network. The latter performs a line feed when you press the RETURN key. Since CDCNET is using the following attribute setting, it also performs a line feed after each return.

END_LINE_POSITIONING=LFS

As a result, your data is double-spaced. To remove the extra line, enter the following command:

```
%CHATA ELP=NONE
```

Now your data is single-spaced; only the public data network issues a line feed.

END_OUTPUT_SEQUENCE (EOS)

Purpose Defines the sequence of characters the network appends to each complete line of output from the service. Some microcomputers look for this prompting string after receiving output as a signal to begin sending input.

Format *END_OUTPUT_SEQUENCE=sequence*

sequence	Defines a sequence of 0 through 4 characters. (See appendix B for detailed information about defining characters.)

Remarks When the service indicates that its output has ended, CDCNET sends additional characters to your terminal before expecting input. These characters are defined by END_OUTPUT_SEQUENCE.

Examples You can define END_OUTPUT_SEQUENCE so that you always know when your output ends. To turn END_OUTPUT_SEQUENCE into a prompt, enter the following CHANGE_TERMINAL_ATTRIBUTE command:

```
%CHATA EOS=7
```

With your END_OUTPUT_SEQUENCE set to BEL, your terminal's bell rings after each command has executed.

END_PAGE_ACTION (EPA)

Purpose

Specifies whether the network divides your output into pages. What the network does when the specified PAGE_LENGTH (minus a line) has been displayed depends upon the setting of this attribute. (See the descriptions of the FORM_FEED_SEQUENCE, HOLD_PAGE, and PAGE_LENGTH terminal attributes later in this chapter for additional information. See also chapter 5.)

Format

END_PAGE_ACTION=FFS
END_PAGE_ACTION=NONE

FFS	Uses the setting you specified for FORM_FEED_SEQUENCE.
NONE	Does not take any action.

The default is NONE.

Remarks

- CDCNET performs an END_PAGE_ACTION after the network has displayed PAGE_LENGTH minus one line of data. The last line is reserved for the END_PAGE_ACTION and related information. If PAGE_LENGTH is infinite (PAGE_LENGTH=0), CDCNET does not use END_PAGE_ACTION.

- If you set the following attributes, HOLD_PAGE and the <OVER> prompt occur before the END_PAGE_ACTION.

- You receive page length minus one physical line, whether or not HOLD_PAGE=ON, because a line is needed for the <OVER> prompt and the page turn signal sent to your terminal.

  HOLD_PAGE=ON
  HOLD_PAGE_OVER=ON
  END_PAGE_ACTION=FFS

Examples

Suppose your terminal does not support a paging function. That is, it does not wrap when data fills the screen. You can correct this situation by changing END_PAGE_ACTION to FFS.

```
%CHATA EPA=FFS
```

Each time output reaches PAGE_LENGTH, this entry causes the network to perform the FORM_FEED_SEQUENCE. If you have set this attribute to an appropriate value, the network clears the screen and starts writing again at the top.

END_PARTIAL_CHARACTER (EPC)

Purpose

Specifies the character that causes CDCNET to end and forward a partial transmission to the service (and does not treat the END_PARTIAL_CHARACTER as data). *A service usually does not act on this data until it receives a complete transmission.* END_PARTIAL_CHARACTER also causes CDCNET to do END_PARTIAL_POSITIONING. (See chapter 5 for more information and chapter 4 for a description of the PARTIAL_CHARACTER_FORWARDING connection attribute.)

Format

END_PARTIAL_CHARACTER=character or NUL

character	Specifies an ASCII character. (See appendix B for detailed information about defining characters.)
NUL	Causes the network to deactivate the transmission of partial data.

The default is LF.

Remarks

- The network recognizes this attribute only when INPUT_EDITING_MODE=NORMAL.
- When the following attribute setting applies to your connection, CDCNET uses the END_PARTIAL_CHARACTER to forward input.

  PARTIAL_CHARACTER_FORWARDING=ON

  If it is OFF, CDCNET does not forward partial transmissions until it receives an END_LINE_CHARACTER.
- When CDCNET encounters an END_PARTIAL_CHARACTER in your input (and PARTIAL_CHARACTER_FORWARDING is ON), it does END_PARTIAL_POSITIONING by sending your terminal a sequence. This produces appropriate cursor positioning for the next line of data.

- When CDCNET encounters an END_PARTIAL_CHARACTER in your input following an END_LINE_CHARACTER (and INPUT_EDITING_MODE=NORMAL), it discards the EPC character that immediately *precedes* or *follows* the ELC.

- Deactivate this terminal attribute for a specific connection with the connection attribute PARTIAL_CHARACTER_FORWARDING.

- Normally you change the END_PARTIAL_CHARACTER to avoid conflict with a service use of the character or to use the current character as data.

Examples

If, for some reason, you need to change END_PARTIAL_CHARACTER, enter a CHANGE_TERMINAL_ATTRIBUTE command at your terminal.

```
%CHATA EPC=NUL
```

END_PARTIAL_POSITIONING (EPP)

Purpose

Defines the character string the network sends to your terminal to perform cursor positioning after you enter the character defined by the END_PARTIAL_CHARACTER terminal attribute. The network sends this sequence after receiving an END_PARTIAL_CHARACTER. (See chapter 5 for more information.)

Format

END_PARTIAL_POSITIONING=CRS
END_PARTIAL_POSITIONING=LFS
END_PARTIAL_POSITIONING=CRSLFS
END_PARTIAL_POSITIONING=NONE

CRS	Uses the setting you specified with CARRIAGE_RETURN_SEQUENCE.
LFS	Uses the setting you specified with LINE_FEED_SEQUENCE.
CRSLFS	Uses the settings you specified with both CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE.
NONE	Does not use any character string.

The default is CRS.

Remarks

- If your terminal automatically repositions the cursor where you want it after an END_PARTIAL_CHARACTER, you may want END_PARTIAL_POSITIONING set to NONE.
- If your terminal does not position the cursor where you want it after an END_PARTIAL_CHARACTER, reset END_PARTIAL_POSITIONING. Select the setting that describes what you want the network to do when it receives an END_PARTIAL_CHARACTER in your input.

Examples If your terminal automatically performs both a line feed and return, you can set the END_PARTIAL_POSITIONING to NONE to avoid duplication. To do so, enter the following CHANGE_TERMINAL_ATTRIBUTE command.

```
%EPP=NONE
```

FOLD_LINE (FL)

Purpose

Specifies whether the network folds output lines that exceed the PAGE_WIDTH setting.

When you set this attribute to ON, the portion of the line that exceeds PAGE_WIDTH is continued (displayed or printed) on the next line. (See the descriptions of the CARRIAGE_RETURN_SEQUENCE terminal attribute earlier in this chapter, and the LINE_FEED_SEQUENCE and the PAGE_WIDTH terminal attributes later in this chapter for information on related settings. Also see chapter 5.)

Format

FOLD_LINE=ON
FOLD_LINE=OFF

ON Folds output lines.

OFF Does not fold output lines.

The default is ON.

Remarks

- If a line of output is too long for a screen or printer, some terminals automatically continue the output on the next physical line. However, others overprint on the same physical line. In these cases, changing FOLD_LINE to ON enables CDCNET to position the cursor for you.

  In other words, the network performs a LINE_FEED_SEQUENCE followed by a CARRIAGE_RETURN_SEQUENCE to force remaining output to the left margin of the next line whenever PAGE_WIDTH is reached.

- Whether your terminal wraps output and how FOLD_LINE is set affects the formatting of your data.

Action	FL=ON	FL=OFF
Terminal wraps.	Produces extra blank line.	Produces satisfactory positioning.
Terminal does not wrap.	Produces satisfactory positioning.	Strikes over or discards characters.

- For FOLD_LINE to work properly, you must set PAGE_WIDTH to your terminal's physical page width.

Examples

You are preparing to print a file in which each set of 30 characters identifies a unique item on your microcomputer's printer. You want it printed with a single item on each line so that you can read the information easily. To format your data in this manner, change PAGE_WIDTH to 30 and FOLD_LINE to ON.

```
%CHATA PW=30 FL=ON
```

FORM_FEED_DELAY (FFD)

Purpose

Increases or decreases the amount of idle time after a FORM_FEED_SEQUENCE. (During this time, the network sends your terminal NUL characters.)

If information is sent to your terminal faster than it can position the cursor (or paper), you can avoid losing data by increasing the idle time with FORM_FEED_DELAY. This additional time allows your terminal to completely erase the screen or move to the top of a new page (or form). (See chapter 5 and the description of the FORM_FEED_SEQUENCE terminal attribute later in this chapter for more information.)

NOTE

Form feed delays are typically used for older, nonbuffered devices that do not support X-ON/X-OFF flow control.

Format

FORM_FEED_DELAY=integer

integer	Specifies the idle time in number of milliseconds. You can use any integer from 0 through 3000.

The default is 0.

Remarks

- Whenever any of the following occur, CDCNET inserts enough NUL characters in the output to fill the milliseconds you specify with FORM_FEED_DELAY:
  - Output contains a FORM_FEED_SEQUENCE, as specified by END_PAGE_ACTION.
  - Output from your service contains a character (such as a format effector) that the network converts to the FORM_FEED_SEQUENCE.

- A combination of factors determines the delay you enter for FORM_FEED_DELAY. The amount of time needed depends on the page length, the speed at which your terminal processes a FORM_FEED_SEQUENCE, and line speed.

Factor	Delay Time
Device processes FFS faster.	Decreases.
Page length increases.	Increases.
Line speed increases.	Decreases.

- Certain terminals, such as the Lear Siegler ADM3A and ADM5, do not ignore NUL characters. In these cases, you should set FORM_FEED_DELAY to 0.

Examples

Slave printers that repeat information exchanged by CDCNET and a terminal are often attached to microcomputers. In this situation, you can avoid losing data on your printed listing by changing the FORM_FEED_DELAY.

```
%CHATA FFD=200
```

Specifying a delay of one-fifth second provides additional time for positioning paper. While the network inserts NUL characters after each FORM_FEED_SEQUENCE, your printer moves the paper to the top of the next page.

FORM_FEED_SEQUENCE (FFS)

Purpose Defines the sequence of characters that causes a page break. This sequence, often called a form feed, typically positions the cursor or paper at the top of the next page. (See the description of END_PAGE_ACTION terminal attribute earlier in this chapter for more information. Also see chapter 5.)

Format *FORM_FEED_SEQUENCE=sequence*

sequence Defines a sequence of 0 through 7 characters. (See appendix B for detailed information about defining characters.)

The default is FF.

Remarks

- CDCNET sends a FORM_FEED_SEQUENCE to your terminal whenever the connected service indicates it should do so.
- If the service uses a format effector that requires a top-of-form operation, CDCNET substitutes the sequence defined by FORM_FEED_SEQUENCE.
- If the sequence specified contains CR or LF (ASCII carriage-return or line-feed characters), the network replaces them with the CARRIAGE_RETURN_SEQUENCE or LINE_FEED_SEQUENCE, respectively.
- CDCNET sends the <OVER> prompt to your terminal when it detects an end-of-page condition (a FORM_FEED_SEQUENCE), the printer or screen is not empty, and the terminal attribute HOLD_PAGE_OVER=ON.

Examples Suppose your terminal emulates a DEC VT100 terminal. You want this terminal to clear its screen each time the service requests a top-of-page operation. Enter the following command to change the FORM_FEED_SEQUENCE:

```
%CHATA FFS=(ESC,'[','H',ESC,'[','J')
```

FUNCTION_KEY_CLASS (FKC)

Purpose Specifies the name of the function key class to be used with the current terminal device.

Format *FUNCTION_KEY_CLASS=DEC_VT100*
FUNCTION_KEY_CLASS=DEC_VT100_GOLD
FUNCTION_KEY_CLASS=DEC_VT220
FUNCTION_KEY_CLASS=SUN_160
FUNCTION_KEY_CLASS=NONE

The default is NONE.

Remarks

- The FUNCTION_KEY_CLASS name must match the name of a currently registered Control Data-defined function key class.
- The FUNCTION_KEY_CLASS can be used only when the INPUT_EDITING_MODE=TRANSPARENT and the TRANSPARENT_CHARACTER_MODE=FORWARD or FORWARD_TERMINATE.

Examples When you use the File Editor application with a VT-100 terminal, function keys have to be followed by a return for the application to act on that function. To avoid using the return key you can, before using the editor, change the FUNCTION_KEY_CLASS as shown below.

```
%CHATA FKC=DEC_VT100
```

HOLD_PAGE (HP)

Purpose

Specifies whether data flow stops at the end of each page (or form).

If this attribute is set to ON, data flow stops when PAGE_LENGTH minus one line is reached. It also stops before a FORM_FEED_SEQUENCE. In both cases, you must then press the RETURN key or enter some data to proceed to the next and subsequent pages.

If HOLD_PAGE is set to OFF, output is displayed or printed without interruption. (See the descriptions of the END_PAGE_ACTION and HOLD_PAGE_OVER terminal attributes earlier in this chapter, and the PAGE_LENGTH terminal attribute later in this chapter for information on related settings. See also chapter 5.)

Format

HOLD_PAGE=ON
HOLD_PAGE=OFF

ON Stops data flow and waits for an input response.

OFF Does not stop data flow.

The default is OFF.

Remarks

- The network waits for you to enter the END_LINE_CHARACTER (usually RETURN) when HOLD_PAGE is ON and PAGE_LENGTH is reached. The network also starts a new page when the following occur:

  1. Entering a network command causes the network to place output on a new page. Actually, it is the NETWORK_COMMAND_CHARACTER followed by an END_LINE_CHARACTER that causes your output to resume on the next page. (That is, entering the percent sign (%) and pressing the RETURN key.)

  2. Entering a line of input to the service causes the network to forward the data to the service and display subsequent output on a new page.

- If you have specified the following attribute settings, HOLD_PAGE occurs before END_PAGE_ACTION.

  HOLD_PAGE=ON
  END_PAGE_ACTION=FFS

- When these settings apply and the network sends the terminal a FORM_FEED_SEQUENCE before PAGE_LENGTH is reached, the network stops sending data and waits until you enter an END_LINE_CHARACTER (usually RETURN) before continuing output on the next page.

- Typically, you use HOLD_PAGE to display data, but you can also specify this attribute when printing forms to allow time for positioning the next form.

- CDCNET does not perform page holding when you are using the following setting:

  PAGE_LENGTH=0.

- If PAGE_WIDTH is greater than zero, CDCNET determines when PAGE_LENGTH is reached by counting the number of physical lines it outputs.

- If PAGE_WIDTH is set to zero (PAGE_WIDTH=0), CDCNET determines when PAGE_LENGTH is reached by counting the number of logical lines sent by the service.

Examples

Suppose you are working at a terminal with the following attribute settings and want to display one page at a time:

HOLD_PAGE=OFF
PAGE_LENGTH=24
PAGE_WIDTH=80

You can enter a CHANGE_TERMINAL_ATTRIBUTE command to change the value of HOLD_PAGE:

```
%CHATA HP=ON
```

HOLD_PAGE_OVER (HPO)

Purpose

Specifies whether an <OVER> prompt is displayed at your terminal when a HOLD_PAGE is encountered. This prompt reminds you that more information is available. (See the descriptions of the END_PAGE_ACTION and the HOLD_PAGE terminal attributes earlier in this chapter, and the PAGE_LENGTH terminal attribute later in this chapter for information on related settings. Also see chapter 5.)

Format

HOLD_PAGE_OVER=ON
HOLD_PAGE_OVER=OFF

ON   Displays <OVER> messages.

OFF   Does not display <OVER> messages.

The default is ON.

Remarks

- CDCNET sends you prompting messages when PAGE_LENGTH is greater than zero and the following settings are specified:

  HOLD_PAGE=ON
  HOLD_PAGE_OVER=ON

- CDCNET sends the following prompting message to your terminal at the end of a page.

  ```
  <OVER>
  ```

- CDCNET sends you the prompting message if it detects a FORM_FEED_SEQUENCE even though the printer or screen is not yet empty. See the description of the terminal attribute FORM_FEED_SEQUENCE earlier in this chapter for more information.

  ```
  <OVER>
  ```

Examples

If you are working at a terminal with the following attribute settings, you may want CDCNET to prompt you at the end of a page.

HOLD_PAGE=ON
HOLD_PAGE_OVER=OFF
PAGE_LENGTH=24

Changing the value of HOLD_PAGE_OVER to ON produces an <OVER> prompt on line 24 of every page.

```
%CHATA HPO=ON
```

LINE_FEED_DELAY (LFD)

Purpose

Increases or decreases the idle time after a LINE_FEED_SEQUENCE. (During this time, the network sends your terminal NUL characters.)

If information is sent to your terminal faster than it can position the paper (or cursor), you can avoid losing data by increasing the idle time with LINE_FEED_DELAY. This additional idle time prevents data from being lost while your terminal is moving the paper (or cursor) to the next line. (See chapter 5 and the description of the LINE_FEED_SEQUENCE terminal attribute later in this chapter for more information.)

NOTE

Line feed delays are typically used for older, nonbuffered devices that do not support X-ON/X-OFF flow control.

Format

LINE_FEED_DELAY=integer

integer — Specifies the idle time in number of milliseconds. You can use any integer from 0 through 1000.

The default is 0.

Remarks

- Whenever any of the following occurs, CDCNET inserts enough NUL characters in the output to fill the milliseconds you specify with LINE_FEED_DELAY.
  - Output from your service contains a character (such as a format effector or LF) that the network converts to the LINE_FEED_SEQUENCE.
  - PAGE_WIDTH is reached; either CDCNET or your terminal folds the line (depending upon the setting of FOLD_LINE); and CDCNET inserts a LINE_FEED_SEQUENCE and CARRIAGE_RETURN_SEQUENCE to reposition your cursor.

- The network performs END_LINE_POSITIONING or END_PARTIAL_POSITIONING and, as a result, sends your terminal a LINE_FEED_SEQUENCE.

- A combination of factors determines the delay you enter for LINE_FEED_DELAY. The amount of time needed to move the paper (or cursor) depends on the terminal's page width, the speed at which the device moves, and line speed.

Factor	**Delay**
Page width increases.	Amount of time increases.
Device performs cursor positioning faster.	Amount of time decreases.
Line speed increases.	Amount of time decreases.

- Since line feeds usually occur in conjunction with returns, the time specified for CARRIAGE_RETURN_DELAY often suffices to position the paper (or cursor).

- Using both the LINE_FEED_DELAY and CARRIAGE_RETURN_DELAY provides a combined maximum delay of two seconds for positioning, if needed.

- Certain terminals, such as the Lear Sieglar ADM3A and ADM5, do not ignore NUL characters. In these cases, you should set LINE_FEED_DELAY to 0.

Examples

Slave printers that repeat information exchanged by CDCNET and a terminal are often attached to microcomputers. To avoid losing data on your printed listing, change the LINE_FEED_DELAY.

```
%CHATA LFD=200
```

Specifying a delay of one-fifth second provides additional time for positioning the paper. While CDCNET inserts NUL characters after each LINE_FEED_SEQUENCE, your printer moves the paper to the next line.

LINE_FEED_SEQUENCE (LFS)

Purpose Defines the sequence of characters that indicates a line-feed action. This action moves the cursor down one line, or rolls the printer paper up one line in preparation of the next line of data. (See the descriptions of the END_LINE_POSITIONING and END_PARTIAL_POSITIONING terminal attributes earlier in this chapter for more information. Also see chapter 5.)

Format *LINE_FEED_SEQUENCE=sequence*

sequence Defines a sequence of 0 through 2 characters. (See appendix B for detailed information about defining characters.)

The default is LF.

Remarks

- CDCNET sends the LINE_FEED_SEQUENCE to your terminal whenever a LF (ASCII character) appears in your data.
- When the service uses a format effector that requires a line-feed action, CDCNET substitutes the sequence defined by LINE_FEED_SEQUENCE.
- If FOLD_LINE is ON, CDCNET inserts a LINE_FEED_SEQUENCE when PAGE_WIDTH is reached.

Examples If your terminal automatically performs both a line feed and carriage return, you can set the LINE_FEED_SEQUENCE to NUL to avoid duplication. To do so, enter the following CHANGE_TERMINAL_ATTRIBUTE command.

```
%CHATA LFS=''
```

NETWORK_COMMAND_CHARACTER (NCC)

Purpose Defines the key that precedes CDCNET commands. This manual uses % to represent the NETWORK_COMMAND_CHARACTER.

Format *NETWORK_COMMAND_CHARACTER=character*

character Specifies an ASCII character. (See appendix B for detailed information about defining characters.)

The default is %.

Remarks

- When this character occurs at the beginning of an input line, CDCNET interprets the data as one of the following:

  1. A CDCNET terminal user command. (See chapter 2.)

  2. A user interrupt. (See chapter 6.)

  3. Page turn in input. (When the NETWORK_COMMAND_CHARACTER is followed by an END_LINE_CHARACTER, the network repositions the cursor [carriage] at the top of your screen [page]. See the description of the HOLD_PAGE terminal attribute in chapter 5.)

- Using the ACTIVATE_AUTO_RECOGNITION command or changing CODE_SET may change the graphic representation (not the value) of your NETWORK_COMMAND_CHARACTER and ATTENTION_CHARACTER.

Examples

Suppose you have a Bubble APL terminal that you normally use with the coded character set switch on ASCII. To use the NOS APL interpreter, you need to change the NETWORK_COMMAND_CHARACTER and ATTENTION_CHARACTER, as well as the switch setting and CODE_SET.

1. Enter the following CHANGE_TERMINAL_ATTRIBUTE command:

   ```
   %CHATA NCC=ESC AC=DC4 CS=TPAPL
   ```

   You change the NETWORK_COMMAND_CHARACTER to avoid having the division sign become your NETWORK_COMMAND_CHARACTER when you change to TPAPL.

2. Set your terminal's switch on APL to select the typewriter-pairing ASCII coded character set with APL print.

When you have completed your APL work and want to use ASCII again, repeat the steps.

1. Enter the command:

   ```
   ESC CHATA NCC=% AC=NUL CS=ASCII
   ```

2. Set your terminal's switch on ASCII.

PAGE_LENGTH (PL)

Purpose Specifies how the network divides output into pages. In conjunction with other attributes, it enables you to work with shorter or longer pages of information. (See the descriptions of the END_PAGE_ACTION, FORM_FEED_SEQUENCE, and HOLD_PAGE terminal attributes earlier in this chapter for information on related settings. Also see chapter 5.)

Format *PAGE_LENGTH=integer*

integer — Specifies the number of lines per page. You can use any integer from 2 through 255. If you enter 0, the network does not divide your output into pages.

The default is 24.

Remarks

- The network informs your service of any change in PAGE_LENGTH.
- PAGE_LENGTH establishes the vertical boundary for your output. To ensure proper paging, you should set PAGE_LENGTH to the actual physical length of your screen or page.
- When PAGE_LENGTH is set to infinite (PAGE_LENGTH=0), the network does not increment a line counter or perform any functions dependent on PAGE_LENGTH, END_PAGE_ACTION, or HOLD_PAGE because you do not have a specified page length. Instead, your output is continuously displayed or printed, regardless of length.
- If a service uses the current page length value to calculate transmission block size or otherwise format output, setting PAGE_LENGTH to infinite (PAGE_LENGTH=0) may produce undesirable results.
- Setting PAGE_LENGTH to a value larger than your terminal's actual screen length can cause the first lines of some displays to scroll off the page during output.

Examples

Some terminals support both 24-line and 30-line lengths. When you are working at one of these terminals, you can avoid any confusion by specifying PAGE_LENGTH after you connect to CDCNET.

```
%CHATA PL=24
```

Issuing this command ensures that PAGE_LENGTH is 24, regardless of which option was previously selected.

PAGE_WIDTH (PW)

Purpose

Determines the page width for your screen or page. (See chapter 5 and the description of the FOLD_LINE terminal attribute earlier in this chapter.)

Format

PAGE_WIDTH=integer

integer — Specifies the number of characters per line. You can use any integer from 10 through 255. If you enter 0, the network does not divide your output into lines.

The default is 80.

Remarks

- The network informs your service of any change in PAGE_WIDTH.
- If PAGE_WIDTH is greater than your terminal's page width, characters may be truncated, overstruck on the right margin, replaced on the left margin of the same line, or continued on the next line depending upon your terminal.

  In addition, insertion of any NUL characters required by CARRIAGE_RETURN_DELAY or LINE_FEED_DELAY does not necessarily coincide with carriage (or cursor) movement.

Examples

If you are using a CDC 721 terminal and have a file containing records greater than 80 characters in length, you can extend the terminal's page width to 132 characters. First, change the terminal's physical setting.

1. Select SETUP to begin the change.
2. Press F10 to specify MORE SELECT.
3. When the new function selections appear on your screen, press SHIFT F7 to increase your terminal's width to 132 characters.
4. Press F1 to return to CYBER mode.

Then, enter the following CHANGE_TERMINAL_ATTRIBUTE command to inform the network of the change by setting PAGE_WIDTH to 132.

```
%CHATA PW=132
```

PARITY (P)

Purpose Tells the network what your terminal's parity is.

If you receive garbled (unreadable) messages when accessing the network, the communication line that connects your terminal to the computer site probably requires a different parity setting (or coded character set). At this point, you have two choices:

1. Call your site administrator to determine your terminal's proper setting. Then enter a CHANGE_TERMINAL_ATTRIBUTE command with your terminal's setting for PARITY (or CODE_SET).

2. Turn off your terminal and consult your terminal documentation to verify the setting.

If necessary, reset your terminal's switch to match the characteristics of your communication line.

Format *PARITY=EVEN*
PARITY=ODD
PARITY=MARK
PARITY=ZERO
PARITY=NONE

EVEN	Specifies that the sum of all bits in a character is an even number.
ODD	Specifies that the sum of all bits in a character is an odd number.
MARK	Sets parity bit to 1. (Certain terminals and software packages require this setting.)
ZERO	Sets parity bit to 0. (Certain terminals and software packages require this setting.)
NONE	Passes parity bit through unchanged.

The default is EVEN.

Remarks

- If your site administrator has configured your communication line for automatic recognition, the two line feed characters the network sends to your terminal in response to the initial returns you use to connect to CDCNET have their parity bits set to zero.

- The following chart shows how CDCNET handles input and output for the various PARITY settings. (The INPUT_EDITING_MODE setting is only relevant when PARITY is NONE.) It describes how CDCNET handles data.

PARITY	Input	Output
EVEN	Checks and clears 8th bit.	Sets 8th bit to EVEN.
ODD	Checks and clears 8th bit.	Sets 8th bit to ODD.
MARK	Checks and clears 8th bit.	Sets 8th bit.
ZERO	Checks and clears 8th bit.	Clears 8th bit.
NONE	Clears 8th bit if input editing mode is normal. Passes through unchanged if input editing mode is transparent.	Clears 8th bit if output is not transparent. Passes through unchanged if output is transparent.

- If CDCNET detects parity errors in your input data, the network forwards the data with an error indication. It is up to your service to process this indication.

- The following attribute settings are often used during file transfers.

  PARITY=NONE
  INPUT_EDITING_MODE=TRANSPARENT

  Instead of checking the 8th bit, file transfer packages use other methods such as CRC.

- You normally change PARITY due to one of the following:
  - The initial value is incorrect for your terminal.
  - The 8th bit in the code must be treated as part of your data.
- You may need to set PARITY to NONE if a local area network is between your terminal and CDCNET. Consult your site administrator for specific information.

Examples

A local area network that links your personal computer to CDCNET requires a connected terminal to have parity set to none. Your site administrator has configured your communication line with PARITY set to EVEN (the default). To avoid problems, you should change the PARITY setting so that CDCNET matches the local area network, which should match your terminal's setting.

```
%CHATA P=NONE
```

With this setting, CDCNET ignores your terminal's parity setting.

RESPONSE_ACTION (RA)

Purpose

Indicates whether the network displays successful network command responses, network banners, and other unsolicited, network-generated output at your terminal.

Format

RESPONSE_ACTION=SEND
RESPONSE_ACTION=DISCARD

SEND	Displays all network responses at terminal.
DISCARD	Discards all network responses.

The default is SEND.

Remarks

- Specifying DISCARD for the RESPONSE_ACTION setting results in discarding network command responses, network banners, and other unsolicited, network-generated output, instead of displaying it at your terminal. This setting is typically used for passthrough connections to non-Control Data hosts or non-Control Data host front end computers.
- Typically, the type of user receiving the information determines the RESPONSE_ACTION setting.

Setting	Type of User
DISCARD	Nonhuman
SEND	Human

Examples

When an asynchronous printer is connected to the NOS Printer Support Utility (PSU), the terminal user procedure establishes attribute settings for the printer. This procedure contains a CHANGE_TERMINAL_ATTRIBUTE command to suppress the CDCNET banner and other unsolicited output.

```
CHANGE_TERMINAL_ATTRIBUTE RA=DISCARD
```

If for some reason you do need to change RESPONSE_ACTION at your terminal, enter a CHANGE_TERMINAL_ATTRIBUTE command.

```
%CHATA RA=DISCARD
```

STATUS_ACTION (SA)

Purpose Specifies how your terminal handles messages from the network operator.

Format *STATUS_ACTION=DISCARD*
STATUS_ACTION=HOLD
STATUS_ACTION=SEND

DISCARD (D)	Does not display messages. CDCNET discards the messages.
HOLD (H)	Saves the last four messages until you change STATUS_ACTION to SEND or DISCARD. If you disconnect from CDCNET, the network discards the messages.
SEND (S)	Displays each message when received.

The default is SEND.

Remarks Unless you specify one of the following options, messages from the network operator may appear in the midst of your output.

STATUS_ACTION=HOLD
STATUS_ACTION=DISCARD

Examples You are preparing to print some financial forms at your terminal. Because you do not want unsolicited messages on the forms, change STATUS_ACTION to HOLD.

```
%CHATA SA=HOLD
```

This CHANGE_TERMINAL_ATTRIBUTE command instructs CDCNET to suppress any messages from the network operator.

TERMINAL_MODEL (TM)

Purpose Identifies your terminal's characteristics for the service you are accessing. If your computer site uses this feature, your site administrator gives you a terminal model name. (See appendix E for detailed information on using TERMINAL_MODEL.)

Format *TERMINAL_MODEL=name*

name	Specifies the name assigned to your terminal model. (See appendix E for a list of valid NOS/VE and NOS names.)

The default is NONE.

Remarks

- The network informs your service of any change in TERMINAL_MODEL.
- If you or your site administrator does not specify TERMINAL_MODEL, it defaults to NONE.

Examples To change your terminal model name from NONE to DEC_VT100, enter the following command:

```
%CHATA TM=DEC_VT100
```

Connection Attributes 4

Connection Attributes 4

Connection attributes enable you to control the connection between your terminal and a service. As mentioned earlier, you can create and use more than one connection to link your terminal with a service. Each connection you create has one set of connection attributes.

Connection attributes generally describe how an application wants to use the terminal. Connection attributes are changed mostly by applications although there may be exceptions where it is meaningful for you to select certain values for connection attributes. Any time you change a connection attribute, CDCNET notifies the connected service of the change.

Connection attributes share the following qualities:

- The proper connection attribute value is set by the application, so you should normally not have to set the value.
- The connection attribute value is changed using commands available to your application through your service.

Displaying and Changing Connection Attributes

Two commands allow you to display and change the connection attributes associated with your terminal. That is, you can display and change the ones supported by the protocol you are using. The network maintains a separate set of connection attributes for each connection you create.

DISPLAY_CONNECTION_ATTRIBUTE

CHANGE_CONNECTION_ATTRIBUTE

Because applications generally set these attributes, most users can begin processing on their connected service without changing any connection attributes. However, you may want to change the settings of ATTENTION_CHARACTER_ACTION or BREAK_KEY_ACTION, depending on your service.

Connection Attributes and Protocols

Figure 4-1 shows the default connection attribute values for the Asynchronous protocol. Default connection attributes for the 3270 Binary Synchronous Communications, X.PC, Mode 4, and 3270 SNA Communications protocols are described in appendixes F, H, I, and J respectively.

```
Attention_Character_Action       : 2
Attention_Character_Enable       : ON
Break_Key_Action                 : 0
Echo_Enable                      : ON
Input_Block_Size                 : 160
Input_Editing_Mode               : NORMAL
Input_Flow_Control_Enable        : ON
Input_Output_Mode                : UNSOLICITED
Output_Flow_Control_Enable       : ON
Parity_Enable                    : ON
Partial_Character_Forwarding     : OFF
Store_Backspace_Character        : OFF
Store_Nuls_Dels                  : OFF
Transparent_Character_Mode       : TERMINATE
Transparent_Forward_Character    : CR 8D(16)
Transparent_Length_Mode          : NONE
Transparent_Message_Length       : 255
Transparent_Protocol_Mode        : NONE
Transparent_Terminate_Character  : CR 8D(16)
Transparent_Timeout_Interval     : 0
Transparent_Timeout_Mode         : NONE
```

Figure 4-1. Asynchronous Protocol Connection Attribute Defaults

CDCNET recognizes the following connection attributes only when INPUT_EDITING_MODE=TRANSPARENT:

- Transparent_Character_Mode
- Transparent_Forward_Character
- Transparent_Length_Mode
- Transparent_Message_Length
- Transparent_Protocol_Mode
- Transparent_Terminate_Character
- Transparent_Timeout_Interval
- Transparent_Timeout_Mode

Connection Attributes and Input Editing Modes

Which attributes apply to your input and output processing depends upon whether the INPUT_EDITING_MODE connection attribute is set to NORMAL or TRANSPARENT, as shown in table 4-1. Chapter 5 describes normal and transparent modes in more detail.

Table 4-1. Connection Attributes and Input Editing Modes

Connection Attribute	Normal	Transparent
Attention_Character_Action	I	I
Attention_Character_Enable		I
Break_Key_Action	I	I
Echo_Enable		I
Input_Block_Size	I	I
Input_Editing_Mode[1]		
Input_Flow_Control_Enable		O
Input_Output_Mode	I/O	I/O
Output_Flow_Control_Enable		I
Parity_Enable		I
Partial_Character_Forwarding	I	
Store_Backspace_Character	I	
Store_Nuls_Dels	I	
Transparent_Character_Mode		I
Transparent_Forward_Character		I
Transparent_Length_Mode		I
Transparent_Message_Length		I
Transparent_Protocol_Mode		I
Transparent_Terminate_Character		I
Transparent_Timeout_Interval		I
Transparent_Timeout_Mode		I

Attributes marked I apply to input. Those marked O apply to output.

1. Table 4-1 is based on the value of this attribute.

Connection Attribute Descriptions

The following pages describe the connection attributes used by the network to control Asynchronous protocol connections. Refer to appendixes F through I for connection attribute support for other protocols.

These attribute descriptions contain purpose, format, remarks, and examples, when applicable. Each description begins with an attribute name followed by its abbreviation, in parentheses. The format description lists the values that you can assign a specified connection attribute. It also gives an explanation of each value and its abbreviation.

ATTENTION_CHARACTER_ACTION (ACA)

Purpose

Specifies how the network responds when it recognizes the ATTENTION_CHARACTER in data from your terminal. (Chapter 3 describes the ATTENTION_CHARACTER terminal attribute.)

Format

ATTENTION_CHARACTER_ACTION=integer

integer — Specifies a service-defined action. You can use any integer from 0 through 9. On NOS/VE and NOS, 0 cancels input, 1 cancels input and output (usually a pause condition on most services), and 2 through 9 cancels input and output (usually a terminate condition on most services).

Default is 2.

Remarks

- ATTENTION_CHARACTER_ACTION and BREAK_KEY_ACTION apply to both normal and transparent modes.
- Whenever you press CONTROL T, the key that you specified for ATTENTION_CHARACTER, CDCNET performs these actions:
  - Sends the service expedited data (the value that you specified for ATTENTION_CHARACTER_ACTION).
  - Discards both input and output.
  - In addition, the service gives control to the program if your program is able to invoke reprieve processing.

Examples

If you are writing a program, you can use ATTENTION_CHARACTER_ACTION to provide users with specialized interrupts. For instance, using the following commands allows users to interrupt your program.

```
%CHACA ACA=7
%CHATA AC=20
```

ATTENTION_CHARACTER_ENABLE (ACE)

Purpose Specifies whether or not the terminal interface program (TIP) is to treat the attention character as data. The parameter is only in effect when the connection attribute INPUT_EDITING_MODE=TRANSPARENT.

Format *ATTENTION_CHARACTER_ENABLE=YES*
ATTENTION_CHARACTER_ENABLE=NO

YES The TIP does attention character processing as specified by the terminal attribute ATTENTION_CHARACTER. If AC=some character, the specified character is processed as the attention character. If AC=NUL, all characters are processed as data, regardless of the value of this (ACE) parameter.

NO The TIP treats the attention character as data.

The default is YES.

Remarks ATTENTION_CHARACTER_ENABLE applies only when INPUT_EDITING_MODE=TRANSPARENT.

BREAK_KEY_ACTION (BKA)

Purpose

Specifies how the network responds when it recognizes a BREAK signal from your terminal. Because the ability to transmit this signal is terminal-dependent and is not defined by a terminal attribute, you should check your terminal documentation for more information.

Format

BREAK_KEY_ACTION=integer

integer	Specifies a service-defined action. You can use any integer from 0 through 9. On NOS/VE and NOS, 0 cancels input, 1 cancels input and output (usually a pause condition on most services), and 2 through 9 cancels input and output (usually a terminate condition on most services).

The default is 0.

Remarks

- ATTENTION_CHARACTER_ACTION and BREAK_KEY_ACTION apply to both NORMAL and TRANSPARENT modes.
- You can have interrupt and terminate keys on most terminals by using your ATTENTION_CHARACTER and BREAK key and by setting the appropriate connection attributes: ATTENTION_CHARACTER_ACTION and BREAK_KEY_ACTION.
- The BREAK signal also acts as an X-ON when connection attribute OUTPUT_FLOW_CONTROL_ENABLE=YES. (See the attribute description later in this chapter.)

Examples

You can use your BREAK key to terminate a program. To do this, change the value of BREAK_KEY_ACTION by entering the following command:

```
%CHACA BKA=2
```

The next time you use the BREAK key while output is in process, the service reacts as if you entered a terminal user interrupt (such as %2). That is, the network discards any input and output, and then the service terminates the program.

ECHO_ENABLE (EE)

Purpose Specifies whether the network echoes characters you enter when the terminal attribute ECHOPLEX=TRUE. (See the description of the ECHOPLEX terminal attribute in chapter 3 for more information.)

Format *ECHO_ENABLE=YES*
ECHO_ENABLE=NO

YES Specifies that the network echoes all characters you enter if terminal attribute ECHOPLEX=TRUE.

NO Specifies that the network does not echo characters you enter if terminal attribute ECHOPLEX=TRUE.

The default is YES.

NOTE

When terminal attribute ECHOPLEX=FALSE, ECHO_ ENABLE is ignored and no echoing occurs.

Remarks None.

INPUT_BLOCK_SIZE (IBS)

Purpose Specifies the maximum number of characters the network forwards to your service in a transmission. Whenever your terminal sends the specified number of characters to the network, it can send the service a partial transmission without an END_LINE_CHARACTER or END_PARTIAL_CHARACTER.

Format *INPUT_BLOCK_SIZE=integer*

integer Specifies the number of characters. You can use any integer from 80 through 2000.

The default is 160.

Remarks

- When a service cannot store transmissions larger than a certain size, you can reduce the size of the network transmissions with INPUT_BLOCK_SIZE.

- The settings of the following attributes determine when the network sends a service transmission. (See chapter 5 for details.)

Type of Data	Attribute
Normal mode	END_LINE_CHARACTER END_PARTIAL_CHARACTER
Transparent mode	TRANSPARENT_CHARACTER_MODE TRANSPARENT_LENGTH_MODE TRANSPARENT_TIMEOUT_MODE

Upon recognizing a condition specified by one of these attributes, the network sends the service the characters that it has received from the terminal.

If the number of characters the network is storing reaches the INPUT_BLOCK_SIZE before one of the above conditions occur, the network forwards the data as a special type of partial transmission.

INPUT_EDITING_MODE (IEM)

Purpose

Specifies how the network edits data you enter at your terminal. When you set this attribute to NORMAL, the network edits information and responds to all CDCNET commands. When set to TRANSPARENT, the network forwards information with only limited editing. In transparent mode, the network only responds to the ATTENTION_CHARACTER, BREAK key, and X-ON/X-OFF characters. (See chapter 5 for additional information. Chapter 3 describes the ATTENTION_CHARACTER terminal attribute.)

Format

INPUT_EDITING_MODE=NORMAL
INPUT_EDITING_MODE=TRANSPARENT

NORMAL (N)	Edits your input before forwarding it to the service. This editing removes special codes or characters used by your terminal.
TRANSPARENT (T)	Forwards your input to the service. The network does not remove special codes or characters.

The default is NORMAL.

Remarks

- Normal mode is used for line-oriented, interactive, command/response processing.
- Transparent mode is frequently used in file transfers, full-screen formatting, and full-screen editing.
- Normally, your service controls the IEM attribute.
- The $NET connection is always in normal mode.

INPUT_FLOW_CONTROL_ENABLE (IFCE)

Purpose Specifies whether or not input character flow control is to be used on the terminal. (See the description of the CHARACTER_FLOW_CONTROL terminal attribute in chapter 3 for more information.)

Format *INPUT_FLOW_CONTROL_ENABLE=YES*
INPUT_FLOW_CONTROL_ENABLE=NO

YES Specifies that the terminal controls the flow of input characters when connection attribute INPUT_EDITING_MODE=TRANSPARENT and terminal attribute CHARACTER_FLOW_CONTROL=INPUT or ON.

NO Specifies that the terminal does not control the flow of input characters when connection attribute INPUT_EDITING_MODE=TRANSPARENT and terminal attribute CHARACTER_FLOW_CONTROL=INPUT or ON.

The default is YES.

Remarks None.

INPUT_OUTPUT_MODE (IOM)

Purpose Specifies how terminal input and service output are handled by the network before delivery.

Format *INPUT_OUTPUT_MODE=FULLDUPLEX*
INPUT_OUTPUT_MODE=SOLICITED
INPUT_OUTPUT_MODE=UNSOLICITED

FULLDUPLEX (F)	Indicates that the network edits and forwards input even if you are receiving output.
SOLICITED (S)	Indicates that the service must request input. The network does not edit or forward input until requested by the service (that is, the network allows you to enter data, but does not perform cursor positioning until the service accepts it).
UNSOLICITED (U)	Indicates that input has priority over output. The network edits input, forwards input, and delays any output that may be in progress (that is, the network allows you to type ahead).

The default is UNSOLICITED.

Remarks

- In fullduplex mode, the network processes input and output independently of each other. It edits input upon receipt, and when it receives a forwarding character, such as an END_LINE_CHARACTER, the network forwards any edited input to the service.

  As soon as the network receives output, it delivers the output to your terminal.

- In solicited mode, the network edits and forwards input only when the service solicits input. The service does this by sending a complete output transmission to the terminal.

If the service does not solicit input, the network accepts input, but does not edit this input. (That is, the network does not echo, backspace, or perform cursor positioning.) While you can enter interrupts, they are not processed until the service solicits input.

The network transmits output upon receipt. Input does not interrupt output in progress.

- In unsolicited mode, the network edits input upon receipt. Edited input is forwarded to the service when a forwarding character, such as an END_LINE_CHARACTER is received.

  When editing input, the network does not send any output received until it encounters a forwarding character. Upon receipt of input, the network suspends any output in progress until it receives a forwarding character.

  If you are using the X.25 Asynchronous protocol, input does not interrupt output. To do so, you can press the BREAK key or issue an X-OFF character if CHARACTER_FLOW_CONTROL is ON.

OUTPUT_FLOW_CONTROL_ENABLE (OFCE)

Purpose Specifies whether or not the output character flow is to be accepted from the terminal when INPUT_EDITING_MODE=TRANSPARENT. (See the description of the CHARACTER_FLOW_ CONTROL terminal attribute in chapter 3 for more information.)

Format *OUTPUT_FLOW_CONTROL_ENABLE=YES*
OUTPUT_FLOW_CONTROL_ENABLE=NO

YES Controls the output flow of data when terminal attribute CHARACTER_FLOW_CONTROL=OUTPUT or ON, and INPUT_EDITING_MODE=TRANSPARENT.

NO Does not take any action with the output flow of data from the terminal when CHARACTER_FLOW_CONTROL=OUTPUT and INPUT_EDITING_MODE=TRANSPARENT.

The default is YES.

Remarks None.

PARITY_ENABLE (PE)

Purpose Specifies whether or not the network processes data according to the setting of the terminal attribute PARITY. (See the description of the terminal attribute PARITY in chapter 3 for more information.)

Format *PARITY_ENABLE=YES*
PARITY_ENABLE=NO

YES Specifies that input and output data are processed according to the setting of the terminal attribute PARITY.

NO Specifies that input and output data are processed as if the terminal attribute PARITY=NONE.

The default is YES.

Remarks The attribute affects only transparent output data. The attribute is in effect for input only when INPUT_EDITING_MODE=TRANSPARENT.

PARTIAL_CHARACTER_FORWARDING (PCF)

Purpose Specifies whether the network forwards a partial transmission when an END_PARTIAL_CHARACTER occurs. (Chapter 3 describes the END_PARTIAL_CHARACTER terminal attribute.)

Format *PARTIAL_CHARACTER_FORWARDING=ON*
PARTIAL_CHARACTER_FORWARDING=OFF

ON	Forwards partial transmission to the service when an END_PARTIAL_CHARACTER occurs in input.
OFF	Does not take any action when an END_PARTIAL_CHARACTER occurs in input.

The default is OFF.

Remarks None.

STORE_BACKSPACE_CHARACTER (SBC)

Purpose Specifies how the network handles the BACKSPACE_CHARACTER in your input data. (Chapter 3 describes the BACKSPACE_CHARACTER terminal attribute.)

Format *STORE_BACKSPACE_CHARACTER=ON*
STORE_BACKSPACE_CHARACTER=OFF

ON Forwards the BACKSPACE_CHARACTER and the character you backspaced over to the service as part of NORMAL data.

OFF Discards BACKSPACE_CHARACTER and the character you backspaced over from NORMAL data.

The default is OFF.

Remarks None.

STORE_NULS_DELS (SND)

Purpose Specifies how the network handles NUL (null) and DEL (delete) characters.

Format STORE_NULS_DELS=ON
STORE_NULS_DELS=OFF

ON	Forwards NUL (null) and DEL (delete) characters to the service as part of NORMAL data.
OFF	Discards NUL and DEL characters from NORMAL data.

The default is OFF.

Remarks Services usually do not consider NUL or DEL characters to be data. Both are normally nonprinting characters that are used by your terminal or the network; most services require NUL or DEL characters to be discarded.

TRANSPARENT_CHARACTER_MODE (TCM)

Purpose

Forwards input data and/or terminates TRANSPARENT mode when data entered from your terminal contains a character specified by TRANSPARENT_FORWARD_CHARACTER and/or TRANSPARENT_TERMINATE_CHARACTER. (See the descriptions of TRANSPARENT_FORWARD_CHARACTER and TRANSPARENT_TERMINATE_CHARACTER later in this chapter for more information. Also see chapter 5.)

Format

TRANSPARENT_CHARACTER_MODE=FORWARD
TRANSPARENT_CHARACTER_MODE=TERMINATE
TRANSPARENT_CHARACTER_MODE=FORWARD_TERMINATE
TRANSPARENT_CHARACTER_MODE=NONE

FORWARD (F)	Sends your data to the service when a TRANSPARENT_FORWARD_CHARACTER occurs in input. Transparent mode remains in effect.
TERMINATE (T)	Sends your data to the service and terminates transparent mode when a TRANSPARENT_TERMINATE_CHARACTER occurs in input.
FORWARD_TERMINATE (FT)	Sends your data to the service when a TRANSPARENT_FORWARD_CHARACTER occurs in input; ends transparent mode when a TRANSPARENT_TERMINATE_CHARACTER occurs after a TRANSPARENT_FORWARD_CHARACTER in input.
NONE (N)	Causes no action when these characters occur.

The default is TERMINATE.

Remarks

- CDCNET forwards input data when one of the following settings applies to your connection and CDCNET recognizes a TRANSPARENT_FORWARD_CHARACTER.

  TRANSPARENT_CHARACTER_MODE=FORWARD

  TRANSPARENT_CHARACTER_MODE=FORWARD_TERMINATE

- CDCNET forwards input data and terminates transparent input mode when one of the following settings applies to your connection.

  TRANSPARENT_CHARACTER_MODE=TERMINATE

  TRANSPARENT_CHARACTER_MODE=FORWARD_TERMINATE

TRANSPARENT_FORWARD_CHARACTER (TFC)

Purpose Identifies the key you press to forward transparent input. The network only recognizes this attribute if INPUT_EDITING_MODE=TRANSPARENT and TRANSPARENT_CHARACTER_MODE=FORWARD or FORWARD_TERMINATE, or TRANSPARENT_PROTOCOL_MODE=FORWARD. (See the descriptions of the TRANSPARENT_CHARACTER_MODE and TRANSPARENT_PROTOCOL_MODE attributes in this chapter for more information.)

Format *TRANSPARENT_FORWARD_CHARACTER=list or NUL*

list — Defines a list of 1 through 4 characters. You can then use any of the keys defined by these characters to forward information. (See appendix B for detailed information about defining characters.)

NUL — Specifies no character is used for transparent forwarding.

The default is CR 8D(16).

Remarks

- If either of the following settings applies to your connection, CDCNET acts on the TRANSPARENT_FORWARD_CHARACTER.

  TRANSPARENT_CHARACTER_MODE=FORWARD

  TRANSPARENT_CHARACTER_MODE=FORWARD_TERMINATE

- If you set TRANSPARENT_FORWARD_CHARACTER to DC1 or DC3 with the following settings, CDCNET recognizes the character as a flow control character.

  INPUT_EDITING_MODE=TRANSPARENT
  CHARACTER_FLOW_CONTROL=ON

  Accordingly, DC1 or DC3 does not function as a TRANSPARENT_FORWARD_CHARACTER and does not reach the service as data.

TRANSPARENT_LENGTH_MODE (TLM)

Purpose Forwards input data and/or ends transparent mode when it has received the number of characters specified by TRANSPARENT_MESSAGE_LENGTH. (See the description of the TRANSPARENT_MESSAGE_LENGTH attribute later in this chapter for more information. Also see chapter 5.)

Format *TRANSPARENT_LENGTH_MODE=FORWARD*
TRANSPARENT_LENGTH_MODE=FORWARD_EXACT
TRANSPARENT_LENGTH_MODE=TERMINATE
TRANSPARENT_LENGTH_MODE=NONE

FORWARD (F)	Sends your data to the service when TRANSPARENT_MESSAGE_LENGTH is reached in input. The message may exceed the specified length. Transparent mode remains in effect.
FORWARD_EXACT (FE)	Sends the exact number of characters specified by TRANSPARENT_MESSAGE_LENGTH when the network receives that number from your terminal. Transparent mode remains in effect.
TERMINATE (T)	Sends your data to the service and terminates transparent mode when the TRANSPARENT_MESSAGE_LENGTH is reached in input.
NONE (N)	Takes no action when the TRANSPARENT_MESSAGE_LENGTH is reached.

The default is NONE.

Remarks None.

TRANSPARENT_MESSAGE_LENGTH (TML)

Purpose Specifies the maximum number of characters in data that the network forwards to your service. The network only recognizes this attribute when INPUT_EDITING_MODE is set to TRANSPARENT, and TRANSPARENT_LENGTH_MODE is other than NONE. (See the description of the TRANSPARENT_LENGTH_MODE attribute earlier in this chapter for more information.)

Format *TRANSPARENT_MESSAGE_LENGTH=integer*

integer Specifies the number of characters. You can use any integer from 1 through 32767 (07FFF(16)).

The default is 255.

Remarks

- CDCNET recognizes TRANSPARENT_MESSAGE_LENGTH when TRANSPARENT_LENGTH_MODE is set to FORWARD, FORWARD_EXACT, or TERMINATE.

- CDCNET forwards transmissions and can terminate transparent mode based upon the number of characters received.

TRANSPARENT_PROTOCOL_MODE (TPM)

Purpose Specifies whether or not transparent data is to be forwarded, based upon a TIP-specific protocol event. The network recognizes this attribute only if INPUT_EDITING_MODE=TRANSPARENT. (See the description of the INPUT_EDITING_MODE earlier in this chapter for more information.)

Format *TRANSPARENT_PROTOCOL_MODE=FORWARD*
TRANSPARENT_PROTOCOL_MODE=TERMINATE
TRANSPARENT_PROTOCOL_MODE=NONE

FORWARD (F)	Sends your data to the service when a TIP-specific protocol event occurs.
TERMINATE (T)	Sends your data to the service when a TIP-specific protocol event occurs, and terminates transparent mode (setting the connection attribute INPUT_EDITING_MODE=NORMAL).
NONE (N)	Takes no action based on a TIP-specific protocol event.

The default is NONE.

Remarks None.

TRANSPARENT_TERMINATE_CHARACTER (TTC)

Purpose

Identifies the key you press to terminate and forward input. The network only recognizes this attribute when INPUT_EDITING_MODE is set to TRANSPARENT. (See the description of the TRANSPARENT_CHARACTER_MODE attribute earlier in this chapter for more information.)

Format

TRANSPARENT_TERMINATE_CHARACTER=list

list	Defines a list of 1 through 4 characters. You can then use any of the keys defined by these characters to terminate and forward information. (See appendix B for detailed information about defining characters.)

The default is CR 8D(16).

Remarks

- If either of the following settings applies to your connection, CDCNET acts on the TRANSPARENT_TERMINATE_CHARACTER.

  TRANSPARENT_CHARACTER_MODE=TERMINATE
  TRANSPARENT_CHARACTER_MODE=FORWARD_TERMINATE

- If you set TRANSPARENT_TERMINATE_CHARACTER to DC1 or DC3 with the following settings, CDCNET recognizes the character as a flow control character.

  CHARACTER_FLOW_CONTROL=ON
  INPUT_EDITING_MODE=TRANSPARENT

  Accordingly, neither DC1 nor DC3 function as a TRANSPARENT_TERMINATE_CHARACTER and do not reach the service as data.

- When writing application programs, you must change connection attributes by using commands available to your applications via the service.

TRANSPARENT_TIMEOUT_INTERVAL (TTI)

Purpose Specifies the number of character times which are allowed to elapse between input characters before the network forwards your data to the service. The network only recognizes this attribute if INPUT_EDITING_MODE is set to TRANSPARENT and TRANSPARENT_TIMEOUT_MODE is other than NONE. (See the description of the TRANSPARENT_TIMEOUT_MODE attribute later in this chapter for more information.)

Format *TRANSPARENT_TIMEOUT_INTERVAL=integer*

integer — Specifies the number of character times. You can use the default value of 0 or any integer from 2 through 255 (0FF(16)). If you use 0, use the following chart, which is based on line speed, to determine the allowed elapsed time. To find out the time allotted to your line speed, select the speed less than or equal to your line speed.

Time Limits (Milliseconds)	Line Speed (≤) (Bits per Second)
400	110
300	600
200	1200
150	3600
80	9600
60	19200
40	38400

The default is 0.

If you enter values 2 through 255, the Terminal Interface Program (TIP) calculates the elapsed time (accurate to 10 milliseconds) based on line speed. If the calculated elapsed time is greater than 1 second, the TIP uses 1 second as the value for the elapsed time.

See chapter 5 for more information.

Remarks

- CDCNET recognizes TRANSPARENT_TIMEOUT_INTERVAL when TRANSPARENT_TIMEOUT_MODE is set to FORWARD or TERMINATE.

- CDCNET forwards transmissions and can terminate transparent mode based upon the elapsed time between characters.

- CDCNET uses an equation similar to the following equation to calculate the value of the TTI. For this equation, assume the terminal is running at a line speed of 19200 bps and the elapsed timeout value is to be changed from the default value of 60 milliseconds to approximately 120 milliseconds, for a calculated value of 230.

$$\text{characters} = 120 \text{ msecs} \times \frac{1 \text{ sec}}{1000 \text{ msec}} \times \frac{19200 \text{ bits}}{1 \text{ sec}} \times \frac{1 \text{ character}}{10 \text{ bits}}$$

M03117

TRANSPARENT_TIMEOUT_MODE (TTM)

Purpose Forwards input data and ends transparent mode when the allotted time elapses between characters. The amount of time allotted depends upon your TRANSPARENT_TIMEOUT_INTERVAL attribute and your line speed.

Format *TRANSPARENT_TIMEOUT_MODE=FORWARD*
TRANSPARENT_TIMEOUT_MODE=TERMINATE
TRANSPARENT_TIMEOUT_MODE=NONE

FORWARD (F)	Sends your data to the service when the allotted time elapses without the network receiving a character.
TERMINATE (T)	Terminates transparent mode when the allotted time elapses without the network receiving a character.
NONE (N)	Takes no action based on time between characters.

The default is NONE.

Remarks You can use flow control with TRANSPARENT_TIMEOUT_MODE. If CHARACTER_FLOW_CONTROL=ON, CDCNET stops counting milliseconds when DC3 encountered in input data, and resumes counting when DC1 appears in your data.

CDCNET Terminal Concepts 5

CDCNET Terminal Concepts 5

This chapter discusses the CDCNET terminal concepts of data mode and data type and how they, together with connection and terminal attributes, affect the appearance of different types of data at your terminal.

Data Modes and Data Types

This subsection provides an overview of the data modes and types and the role of attribute settings in each mode.

Data mode specifies how the network processes input and output data. CDCNET recognizes two data modes: virtual line mode and transparent mode. The data mode depends upon the type of data input to the network.

Data type specifies the type of data to be input to the network. CDCNET recognizes two types of data: normal data and transparent data. Normal data is data which is to be edited for input or formatted for output. Normal data is processed in virtual line mode. Transparent data is input data on which little or no editing is to be done. Transparent data is processed in transparent mode. The following table provides more information about data mode and data type.

Type of Data	Description
Virtual line mode	This mode is also referred to as *edited* for input or *formatted* for output. The INPUT_EDITING_MODE connection attribute designates input in this mode as normal. In virtual line mode, the network and the service send each other 7-bit data with the parity bit set to zero or 8-bit data. The network recognizes CDCNET commands when you are processing in this mode. (See chapter 2 for information on commands. To determine which attributes apply when you are entering normal data, see tables 3-1 and 4-1.)
Transparent mode	This mode is called transparent because the network performs little or no editing on input. The INPUT_EDITING_MODE connection attribute designates input in this mode as transparent. In transparent mode, the network and the service can send each other 7-bit data with the parity bit set to zero or 8-bit data. This mode is used by full-screen applications and by file transfer applications to or from microcomputers. For some uses, it is referred to as binary mode. The network does not recognize CDCNET commands when you are processing in transparent mode. In an emergency, you can use the transparent input editing mode escape sequence, the escape to $NET connection sequence, or the secure access sequence to escape from transparent mode. (For more information, see chapter 6. To determine which attributes apply when you are entering transparent data, see tables 3-1 and 4-1.)

Role of Attribute Settings

The role attributes play in determining appearance of data on your terminal varies according to the type of data and protocol you are using. Because you can generally set and change attributes independently of each other, understanding which attributes are involved in certain activities allow you to change or nullify attribute settings more effectively.

Basically, you have full control over the setting of individual attributes, including changing the TIP-defined and site-defined defaults. Any change you make in the subset of attributes that a TIP supports applies for the duration of a connection, or until changed again. In addition, some services may override attribute settings without you knowing about it. (For example, see NOS/VE System Usage for information on how NOS/VE handles connection attributes.)

The rest of this chapter discusses how certain attribute settings affect specific tasks you do at your terminal depending upon the data mode in which you are operating. Depending upon the task, the chapter enumerates your choices or outlines the order in which the network applies attribute settings. (See chapters 3 and 4 for displays showing the attribute defaults for the asynchronous protocol; refer to appendixes F, G, H, I, and J for information on the terminal and connection attributes of other protocols.) This chapter contains the following subsections discussing the attribute settings in virtual line mode and transparent mode:

Subsection	Description
Virtual Line Mode: Data Placement	Identifies the sequences and delays (terminal-attribute settings) that affect the placement of input and output on your screen or page.
Virtual Line Mode: Normal Input	Discusses the use of complete and partial transmissions, related cursor positioning, and data editing.
Virtual Line Mode: Output	Describes formatting virtual line output, end-of-line positioning, and end-of-page positioning.
Transparent Mode: Transparent Input	Discusses editing and forwarding of data.

Subsection	Description
Transparent Mode: Output	Summarizes how the network sends output to your terminal.
Virtual Line and Transparent Modes: Flow Control	Discusses the use of X-ON/X-OFF flow control and EIA flow control.
Virtual Line and Transparent Modes: Interrupts	Describes the use of the BREAK key, ATTENTION_CHARACTER, and user interrupt.

Virtual Line Mode: Data Placement

The network sends your terminal certain sequences, defined by terminal attribute-settings, that affect the appearance of your input and output. What your terminal does with these depends upon how your terminal interprets the characters used in the sequences. As required, you can change these settings to meet your programming aims or to match your terminal's operations.

Some terminology used in these settings, as well as other attributes, originally applied to printers and hardcopy terminals. Now this terminology also applies to video-display terminals. In other words, you use the same terms to describe positioning of a cursor on a screen as paper in a printer. For example in this manual, pressing the RETURN key means you perform a carriage return (send an END_LINE_CHARACTER to the network). On the display terminal, the cursor moves; on the printer, the carriage moves to the left margin.

Using Cursor Positioning Sequences

The following terminal attributes are associated with sequences that affect the placement of the cursor on your screen or printed page. The network sends your terminal a sequence whenever a particular event occurs. It does this by monitoring data that is being transmitted to and from your terminal.

Terminal Attribute	Abbreviation
CARRIAGE_RETURN_SEQUENCE	CRS
END_PAGE_ACTION	EPA
FORM_FEED_SEQUENCE	FFS
LINE_FEED_SEQUENCE	LFS
END_OUTPUT_SEQUENCE	EOS

The network uses the first four sequences to position virtual line data on your screen or printed page. How they affect the appearance of data is described later in this chapter under the following topics.

- Editing NORMAL data (input)
- End-of-line positioning (output)
- End-of-page positioning (output)

END_OUTPUT_SEQUENCE performs end-of-message processing in both virtual line and transparent modes. (Chapter 3 describes this terminal attribute.)

Delaying Data Transmission

If your terminal needs additional time to reposition, the network uses delays established by three terminal attributes to stop the transmission of output so that your terminal or printer can position itself without losing any data. These delays are associated with the following terminal attributes.

Terminal Attribute	Associated Delay (Abbreviation)
CARRIAGE_RETURN_SEQUENCE	CARRIAGE_RETURN_DELAY (CRD)
FORM_FEED_SEQUENCE	FORM_FEED_DELAY (FFD)
LINE_FEED_SEQUENCE	LINE_FEED_DELAY (LFD)

The delays stop transmission of output for a specified number of milliseconds. During this time, the network sends your terminal NUL characters instead of output. What your terminal or printer does with NUL characters during this time depends upon the operation of the individual device. Most terminals discard these characters. If your terminal does not, you should avoid using delays with it. In most cases, terminals operate without requiring any delays.

NOTE

It is usually older, nonbuffered devices that require carriage return, form feed, and line feed delays.

Virtual Line Mode: Normal Input

When the network receives normal input in virtual line mode, the network performs cursor positioning and forwards the data to your service. The following terminal attributes determine how it is forwarded and what the network does to position the cursor. The abbreviations are shown in parentheses.

Terminal Attribute	Associated Character
END_LINE_POSITIONING (ELP)	END_LINE_CHARACTER (ELC)
END_PARTIAL_POSITIONING (EPP)	END_PARTIAL_CHARACTER (EPC)

Your terminal sends the network each character when you enter it. The network stores these characters until it receives an END_LINE_CHARACTER or END_PARTIAL_CHARACTER as specified by the associated terminal attribute. It then forwards the characters to the service as a complete or partial transmission.

Whether you end a line of input with an END_LINE_CHARACTER or END_PARTIAL_CHARACTER (if the PARTIAL_CHARACTER_FORWARDING connection attribute is set to ON) determines if the service receives a complete or partial transmission. In both cases, the network does the following:

- Edits the data before forwarding it.
- Marks a field at the beginning of the transmission indicating whether it is complete or partial.
- Performs some type of cursor positioning.

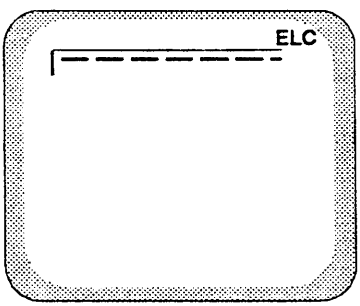
Forwarding Complete Transmissions

Typically, services expect complete transmissions and the network performing END_LINE_POSITIONING. Forwarding a complete transmission begins when you press the RETURN key (or equivalent) and your terminal sends the network an END_LINE_CHARACTER, which is usually defined as CR (ASCII carriage-return character).

Upon receipt of this character, the network does the following:

- Finds the END_LINE_CHARACTER in your data.
- Labels the transmission as complete and forwards it to the service.
- Performs END_LINE_POSITIONING by sending the terminal the sequence specified for this terminal attribute.

How the service handles the complete transmission depends upon the service.

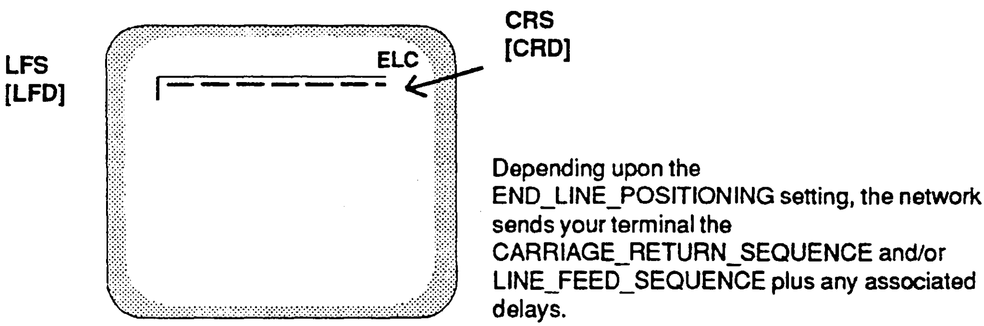


When the network encounters an END_LINE_CHARACTER in your input, it performs END_LINE_POSITIONING.

For example, your cursor is typically repositioned on the left margin of the next line. When you use the Asynchronous protocol, the following defaults are used in END_LINE_POSITIONING.

END_LINE_CHARACTER=CR
END_LINE_POSITIONING=LFS
LINE_FEED_SEQUENCE=LF
LINE_FEED_DELAY=0

When you press the RETURN key, the terminal moves the cursor (or carriage) to the left margin and sends CR (ASCII carriage-return character) to the network as the END_LINE_CHARACTER. In response to this character, the network does END_LINE_POSITIONING by sending the terminal the LINE_FEED_SEQUENCE and a LINE_FEED_DELAY if specified. When the terminal receives LF, it moves the cursor (or carriage) to the next line.



The network also finds the END_LINE_CHARACTER in the data, labels the transmission complete, and forwards the transmission to the service.

Forwarding Partial Transmissions

If your service is able to handle segments of data, it might use the PARTIAL_CHARACTER_FORWARDING connection attribute. Depending upon the service, there can be some overlapping of time in which entering and processing of data can occur. As a result, the service may operate more efficiently.

When you are using the following setting, the network forwards data whenever it receives an END_PARTIAL_CHARACTER in your input.

PARTIAL_CHARACTER_FORWARDING=ON

If PARTIAL_CHARACTER_FORWARDING is OFF, the network does not forward partial transmissions. However, the network forwards a complete transmission whenever an END_LINE_CHARACTER appears in the input, regardless of the PARTIAL_CHARACTER_FORWARDING setting.

Regardless of the PARTIAL_CHARACTER_FORWARDING setting, the network performs END_PARTIAL_POSITIONING when the END_PARTIAL_CHARACTER occurs.

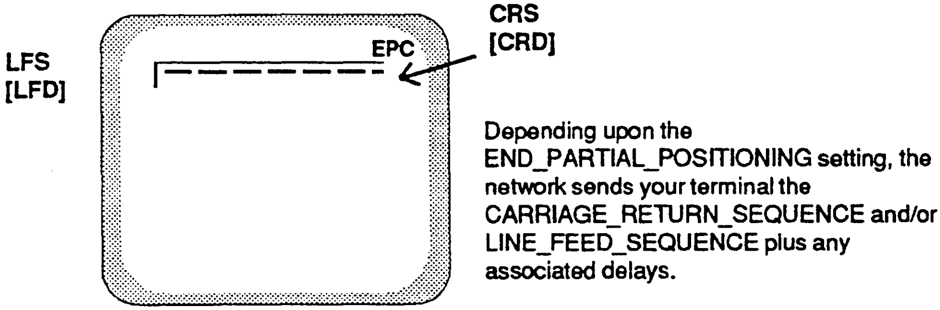
By labeling a transmission partial, the network is telling the service that more data is needed before the transmission is complete. Whether the service waits for the rest of the transmission, goes ahead and processes the partial transmission, or performs some other action, depends upon the service.

Your cursor is generally repositioned in the same place regardless of whether the network performs END_LINE_POSITIONING or END_PARTIAL_POSITIONING. (Differences could be produced by changing terminal-attribute settings.)

The network uses different terminal attributes when performing END_PARTIAL_POSITIONING than it does when performing END_LINE_POSITIONING. With the Asynchronous protocol, the network uses the following defaults.

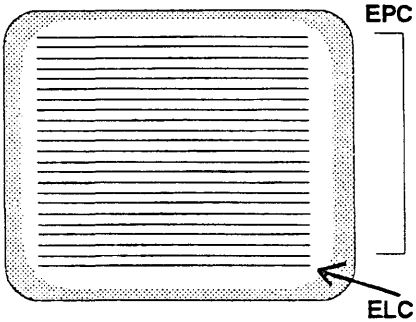
END_PARTIAL_CHARACTER=LF
END_PARTIAL_POSITIONING=CRS
CARRIAGE_RETURN_SEQUENCE=CR
CARRIAGE_RETURN_DELAY=0

For example, when you press the key your terminal designates for sending a partial transmission, the terminal moves the cursor (or carriage) down a line and sends LF (ASCII line-feed character) to the network as the END_PARTIAL_CHARACTER. In response, the network does END_PARTIAL_POSITIONING by sending the terminal the CARRIAGE_RETURN_SEQUENCE (and a CARRIAGE_RETURN_DELAY if specified). When the terminal receives CR, it moves the cursor (or carriage) to the left margin.



The network also finds the END_PARTIAL_CHARACTER in the data, labels the transmission partial, and forwards the transmission to the service.

When the network forwards transmissions to a service, it identifies them as either partial or complete. How the service processes them depends upon the individual service.



When the network sees an END_PARTIAL_CHARACTER on each of these lines, it forwards the partial transmission to the service.

When the network detects the END_LINE_CHARACTER, it indicates to the service that the transmission (which includes previously forwarded partials) is complete .

Editing Normal Data

While editing normal data in virtual line mode, the network looks for the characters listed in the following table. When the network encounters one of these characters, it performs a special action, as summarized in the table and described in detail in chapters 3 and 4.

Character	Action
AC	CDCNET performs the attention character action, as defined by the ATTENTION_CHARACTER_ACTION connection attribute if the connection attribute ATTENTION_CHARACTER_ENABLE is set to ON.
BC	CDCNET backspaces one character over the current input line, but only if the STORE_BACKSPACE_CHARACTER connection attribute is set to OFF.
BLC	CDCNET discards the BLC and recognizes the beginning of a new line of input when this character follows an END_LINE_CHARACTER.
CLC	CDCNET cancels the line being edited when this character is followed by an END_LINE_CHARACTER.
DC1 & DC3	CDCNET recognizes these flow control characters if the terminal attribute CHARACTER_FLOW_CONTROL is set to ON or OUTPUT and OUTPUT_FLOW_CONTROL_ENABLE is set to ON.
ELC	CDCNET recognizes the end of a line of input data when it receives this character.
EPC	CDCNET recognizes the end of a partial line of input data if the PARTIAL_CHARACTER_FORWARDING connection attribute is set to ON and it receives this character.

Character	Action
NCC	When an input line begins with this character and ends with an END_LINE_CHARACTER, CDCNET attempts to execute the input line as a user interrupt or CDCNET terminal user command.
NUL & DEL	CDCNET forwards NULs and DELs to the connected service if the STORE_NULS_AND_DELS connection attribute is set to ON.

Virtual Line Mode: Output

When the network receives virtual line mode output from a service, it formats and translates it as needed. The translating includes converting data to the specified coded character set and changing the formatting characters the service uses to the appropriate network sequences. The following topics in this section describe how terminal attributes affect the appearance of virtual line output on your screen or printed page.

Topic	Description
Formatting virtual line output	Describes the formatting characters and the corresponding network sequences.
End-of-line positioning	Outlines the terminal attributes that may affect the placement of a single line on your screen or page.
End-of-page positioning	Enumerates the terminal attributes that may affect the formatting of an entire screen or page of data.

Formatting Virtual Line Output

When a service sends output to your terminal in virtual line mode, the network expects it to contain certain characters that indicate how the output should be formatted on your screen or page.

Characters	Description
ASCII characters	Virtual line output is composed of 7-bit ASCII characters. These characters are grouped into transmissions of zero or more data characters.
Unit separator	The network expects each transmission displayed or printed at your terminal to end with an ASCII unit-separator character (US).
Format effectors	A service can send virtual line output to the network with format effectors or without format effectors. These characters, which occur in the first position of a line, control the positioning of printed or displayed data.

How output appears on your screen or printed page depends upon the service's use of format effectors and certain ASCII characters. A service can control the appearance of output by doing any of the following:

- Using certain ASCII characters (CR, LF, FF)
- Omitting format effectors
- Using format effectors

Using Certain ASCII Characters

The network translates most characters in virtual line output to the specified coded character set before transmitting them to the terminal. The following three ASCII characters that the network uses to position output are the only exceptions.

ASCII Character	Terminal-Attribute Setting Sent by Network
CR	CARRIAGE_RETURN_SEQUENCE
FF	FORM_FEED_SEQUENCE
LF	LINE_FEED_SEQUENCE

When the network receives one of these characters, it sends your terminal the sequence for the appropriate terminal attribute. As a result, the service is able to control end-of-line and end-of-page positioning through its use of these three characters.

Omitting Format Effectors

If the service does not use format effectors, the network treats each transmission as if it began with a space format effector. In other words, the network sends your terminal a CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE before each line. When the output is displayed or printed, it is single spaced.

Using Format Effectors

In virtual line mode, a service can use format effectors to control the appearance of output on your screen or printed page. In such cases, the network processes the first character of each transmission and the first character after each ASCII unit-separator (US) as a format effector.

Format Effectors

The network converts the format effectors into terminal attribute-defined sequences and sends them to your terminal for line and page positioning.

Table 5-1 lists the format effectors that affect positioning before a line is displayed, and table 5-2 lists those that affect positioning afterwards.

Table 5-1. Positioning Before Output

Format Effector	Terminal Attribute(s) Sent by the Network
+ (plus)	CARRIAGE_RETURN_SEQUENCE
0 (zero)	CARRIAGE_RETURN_SEQUENCE, LINE_FEED_SEQUENCE, and LINE_FEED_SEQUENCE[1]
- (hyphen)	CARRIAGE_RETURN_SEQUENCE, LINE_FEED_SEQUENCE, LINE_FEED_SEQUENCE, and LINE_FEED_SEQUENCE[1]
, (comma)	None
* or 1	FORM_FEED_SEQUENCE
space	CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE[1]
ASCII character not listed in table 5-1 or 5-2	CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE[1]

1. If an input line precedes the output line, the network sends one less LINE_FEED_SEQUENCE to compensate for the positioning performed in response to the preceding input.

Table 5-2. Positioning After Output

Format Effector	Terminal Attribute(s) Sent by the Network
. (period)	CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE
/ (slant)	CARRIAGE_RETURN_SEQUENCE

Example

This example is based on two assumptions:

- The service controls output appearance with format effectors.
- Your terminal is using the following default settings for the Asynchronous protocol:

  CARRIAGE_RETURN_SEQUENCE=CR
  LINE_FEED_SEQUENCE=LF

If the network detects a zero (0) in the first position of a line, it sends the CARRIAGE_RETURN_SEQUENCE and LINE_FEED_SEQUENCE as specified in table 5-1. In other words, it sends your terminal the following ASCII characters to double space the output.

CR
LF
LF

Your terminal interprets these characters and normally positions the cursor at the left margin and performs two line feeds.

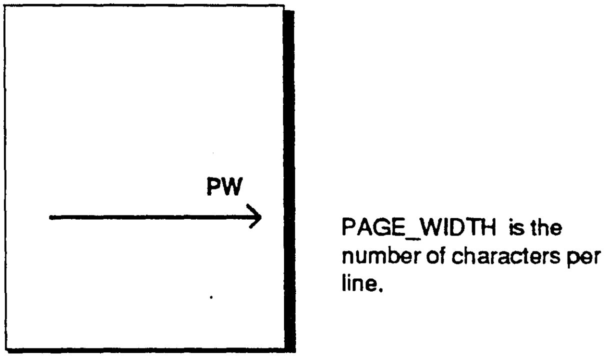
End-of-Line Positioning

The following terminal attributes control how lines of virtual line output appear on your screen or printed page. Abbreviations for these attributes are shown in parentheses.

Terminal Attribute	Description
PAGE_WIDTH (PW)	Determines the maximum number of characters on a line.
FOLD_LINE (FL)	Indicates whether the network performs line folding.
CARRIAGE_RETURN_SEQUENCE (CRS)	Is used in line folding.
CARRIAGE_RETURN_DELAY (CRD)	Is available when CARRIAGE_RETURN_SEQUENCE is sent to your terminal or when the terminal is folding lines that exceed its physical page width.
LINE_FEED_SEQUENCE (LFS)	Is used in line folding.
LINE_FEED_DELAY (LFD)	Is available when LINE_FEED_SEQUENCE is sent to your terminal or when the terminal is folding lines that exceed its physical page width.

Determining PAGE_WIDTH

The setting of PAGE_WIDTH determines the width of an output line. If the virtual lines sent by your service exceed the PAGE_WIDTH setting, the line can be folded so that subsequent output is displayed on the next line.



Using FOLD_LINE

Your terminal may automatically fold lines, in which case FOLD_LINE is usually set to OFF. If it does not, you may want to change the setting. If FOLD_LINE is ON, the network performs line folding as follows:

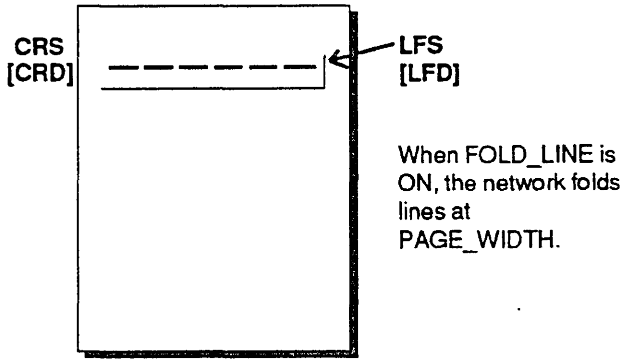
1. The network folds a line if placing a character on a line would exceed the PAGE_WIDTH setting.

2. The network folds the line after placing up to the number of characters specified by PAGE_WIDTH on that line and then performs screen or paper positioning to output the remainder on the next line. The network does this by sending the terminal a LINE_FEED_SEQUENCE followed by a CARRIAGE_RETURN_SEQUENCE.

3. If a CARRIAGE_RETURN_DELAY or LINE_FEED_DELAY is specified to provide additional time for carriage or cursor positioning, the network sends NUL characters after the CARRIAGE_RETURN_SEQUENCE or LINE_FEED_SEQUENCE.

4. Finally, the network displays or prints more output on the next line.

For example, if you use the default settings of the Asynchronous TIP, CARRIAGE_RETURN_SEQUENCE is CR (ASCII carriage-return character) and LINE_FEED_SEQUENCE is LF (ASCII line-feed character). Typically, these settings move your cursor (or carriage) across the page to the beginning of the next line and down a line.



End-of-Page Positioning

The following terminal attributes control how virtual line output appears on your screen or printed page. Abbreviations for these attributes are shown in parentheses.

Terminal Attribute	Description
PAGE_LENGTH (PL)	Determines the maximum number of lines on a page.
HOLD_PAGE (HP)	Indicates whether or not the network stops output at the end of a page and waits for you to press the RETURN key.
HOLD_PAGE_OVER (HPO)	Indicates whether or not the network places an <OVER> message at the end of a page.
END_PAGE_ACTION (EPA)	Indicates whether or not the network uses the FORM_FEED_SEQUENCE to divide your output into pages.
FORM_FEED_SEQUENCE (FFS)	May be specified by END_PAGE_ACTION.
FORM_FEED_DELAY (FFD)	Is available when FORM_FEED_SEQUENCE is sent to your terminal.

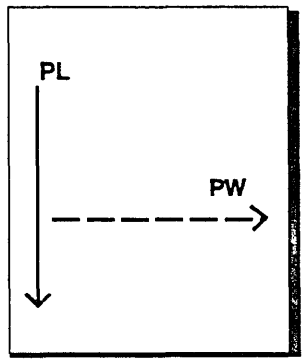
Determining PAGE_LENGTH

While the PAGE_LENGTH setting determines the number of lines on your screen or printed page, the actual number you see is one less than the setting if either of the following conditions exist:

- HOLD_PAGE is set to ON.
- END_PAGE_ACTION is set to FFS and FORM_FEED_SEQUENCE is specified (not blank).

Then the last line of PAGE_LENGTH is reserved for network communications (such as the <OVER> prompt).

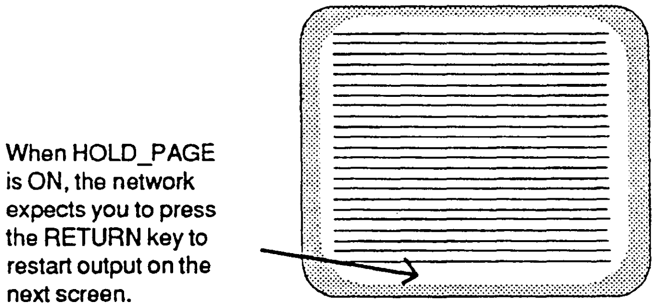
PAGE_LENGTH is the number of lines per page.



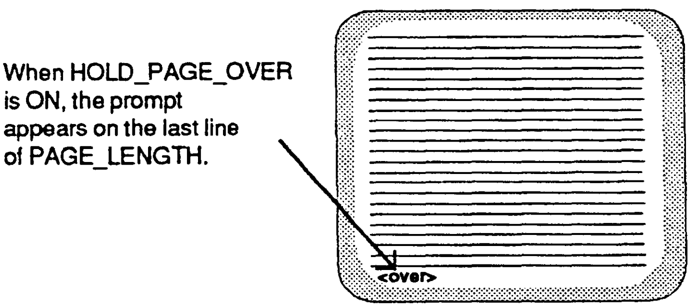
Changing Pages

When the PAGE_LENGTH setting is reached, what happens next depends upon the settings of other terminal attributes.

1. If HOLD_PAGE is ON, the network stops sending output until you indicate you are ready for more. You usually do this by pressing the RETURN key.



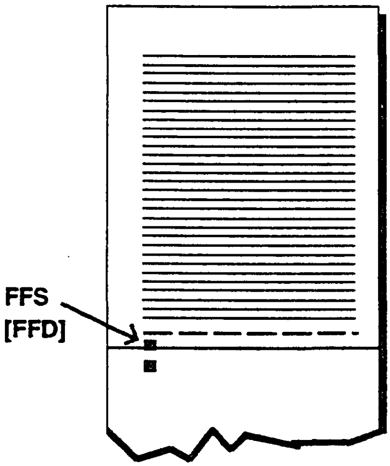
2. If HOLD_PAGE is ON, you can decide whether to receive <OVER> prompts. When HOLD_PAGE_OVER is ON, the network automatically sends a prompt at the end of every page.



3. If END_PAGE_ACTION is set to FFS, the network sends the FORM_FEED_SEQUENCE to your terminal. The network sends this sequence regardless of whether HOLD_PAGE is ON or OFF.

4. If a FORM_FEED_DELAY is specified, the network sends it after the FORM_FEED_SEQUENCE to provide additional time for your terminal to reposition its cursor (or carriage).

For example, if you use the Asynchronous TIP default setting, FORM_FEED_SEQUENCE is FF (ASCII form-feed character). As a result of this setting, your terminal typically positions the cursor at the top of the screen or rolls a new page of paper into place.



The FORM_FEED_SEQUENCE tells your terminal how to position itself for subsequent output.

If a FORM_FEED_DELAY is specified, the network sends it after the FORM_FEED_SEQUENCE to provide additional time for repositioning.

Transparent Mode: Transparent Input

When you enter input data in transparent mode, you use the following connection attribute setting, and the network forwards input data in transparent mode.

INPUT_EDITING_MODE=TRANSPARENT

Editing Transparent Data

In contrast to normal input, the network performs limited editing on transparent data. It only uses the following terminal attributes:

ATTENTION_CHARACTER
CHARACTER_FLOW_CONTROL
ECHOPLEX
END_OUTPUT_SEQUENCE
PARITY
STATUS_ACTION
FUNCTION_KEY_CLASS

The network does not respond to network commands when handling TRANSPARENT input. However, it does recognize the following:

- ATTENTION_CHARACTER
- BREAK key
- DC1 and DC3 characters (if CHARACTER_FLOW_CONTROL is set to ON)

Forwarding Transparent Data

The service usually begins, controls, and ends transparent mode (both input and output) without your intervention. (See chapter 6 for a description of the transparent input editing mode escape sequence.)

Three connection attributes govern the forwarding and terminating of transparent data (input). Each of these depends upon a specific condition being fulfilled.(See chapter 4 for a description of these connection attributes.)

Condition	Connection Attribute
Timeout	TRANSPARENT_TIMEOUT_MODE Identifies the action the network takes after a specified input inter-character time out. The length of this period depends upon your line speed.
Length	TRANSPARENT_LENGTH_MODE Identifies the action the network takes when it has received the number of characters specified by the TRANSPARENT_MESSAGE_LENGTH connection attribute.
Character	TRANSPARENT_CHARACTER_MODE Identifies the action the network takes under the following conditions: • When data entered from your terminal contains a character specified by either the TRANSPARENT_ FORWARD_CHARACTER or TRANSPARENT_ TERMINATE_CHARACTER connection attribute. • As a result of what is specified in the FUNCTION_ KEY_CLASS terminal attribute.

You can use one or a combination of these connection attributes to forward and terminate your data.

Transparent Mode: Output

The network sends transparent output in transparent mode to your terminal unformatted and untranslated. Since it does not perform any end-of-line or end-of-page positioning, the terminal displays data exactly as the service sends it.

If, however, PARITY is EVEN, ODD, MARK, or ZERO, the network ensures that transparent output conforms to the setting. When PARITY is NONE, output is delivered as 8-bit data.

Virtual Line and Transparent Modes: Flow Control

Sometimes your terminal or the network sends virtual line or transparent data faster than the receiver can handle it. To prevent loss of input and output data, the network supports two kinds of automatic flow control:

Type of Flow Control	Description
X-ON/X-OFF	You can control the sending of X-ON/X-OFF characters with the CHARACTER_FLOW_CONTROL terminal attribute. The setting affects the flow of data between the terminal and the network. (See Inadvertent Flow Control in chapter 9 and the description of the CHARACTER_FLOW_CONTROL terminal attribute in chapter 3 for more information.)
EIA (Electronics Industries Association)	The site administrator can set a DEFINE_LINE parameter to control the use of EIA flow control. Setting this parameter to ON or OFF only affects the flow of input data. However, EIA flow control always applies to output. (See Understanding Configuration Options in chapter 1.)

X-ON/X-OFF Flow Control

When CHARACTER_FLOW_CONTROL is set to ON, the X-ON/X-OFF protocol regulates transmission of both input and output data by using ASCII characters. In other words, the terminal and the network control data flow by sending each other DC1 and DC3 characters:

- X-OFF (DC3) turns off transmission.
- X-ON (DC1) turns on transmission.

Automatic Control

The terminal sends DC1 and DC3 characters to control the flow of output being sent by the network. Many terminals automatically send a DC3 character when they cannot handle any more data and then send a DC1 character when they are ready for additional data. The latter means that the terminal has moved most of the data previously received to its screen, a flexible disk, or other destination, and can handle more data.

Similarly, the network sends DC1 and DC3 characters to control the flow of input from the terminal. The network sends a DC3 character when the connection to the service becomes congested, and sends a DC1 character when the connection becomes uncongested.

If either your input or output contains DC1 or DC3 characters that you do want handled as data, you can use EIA flow control instead of CHARACTER_FLOW_CONTROL. If you need to do so, contact your site administrator.

Manual Control

If necessary, you can manually send X-ON/X-OFF characters to control data flow. This manual option is advantageous when you are either using a terminal that does not support automatic flow control, are experiencing communications difficulties, or are scanning a large output file.

- To stop output, press the CONTROL key and hold it down while you press S. This control sequence forces the terminal to send the network an X-OFF character (DC3).

- To restart output, press the CONTROL key and hold it down while you press Q. This control sequence forces the terminal to send the network an X-ON character (DC1).

EIA Flow Control

The EIA_FLOW_CONTROL parameter on the DEFINE_LINE network configuration command determines whether Clear to Send (CTS) and Request to Send (RTS) flow control signals are used by the TDI to stop and resume the flow of the terminal's input data.

When EIA flow control is enabled and the TDI cannot accept terminal input data due to temporary network congestion, the TDI drops its RTS signal, causing the terminal to see its CTS signal drop. A drop of the CTS signal is an indication to the terminal that it should temporarily suspend input transmission. When the TDI can accept input data again it raises its RTS signal, causing the terminal to see its CTS signal raise. This indicates that it is clear to resume transmission of input data.

The terminal can use its RTS signal in the same manner to control the flow of the TDI's output data. Use of EIA flow control by the terminal is always available and is not subject to the EIA_FLOW_CONTROL parameter on the DEFINE_LINE network configuration command.

When both EIA flow control and character flow control are enabled, the TDI uses EIA flow control to control the terminal's input and the TDI recognizes both types of signals as output flow control requests from the terminal.

Although EIA flow control can be used at the same time as CHARACTER_FLOW_CONTROL, there are times when you should only use the former. For example, you should select the EIA option when transferring data using the full 256-character set, and when transmitting data containing DC1 and DC3 characters. If you want to avoid using CHARACTER_FLOW_CONTROL and do not know whether your site controls input with EIA flow control, contact your site administrator.

Terminal User Exception Processing 6

Terminal User Exception Processing 6

Terminal user exception processing allows you to send special signals to your connected service and to CDCNET. CDCNET supports two types of signals, as follows:

- User interrupts (signals to your connected service)
- Special sequences (signals to CDCNET)

User interrupts allow you to send a single character to your connected service.

Special sequences allow you to send specific key/character sequences to CDCNET. CDCNET takes specific actions after recognizing one of these specific sequences.

User Interrupts

Interrupts are control signals sent to your connected service that request the service to temporarily or permanently stop its current activity and begin a new or different activity. This new activity is determined by the value of the character or numeric following the network control character used by your site to identify a user interrupt to the service.

CDCNET processes single-character user interrupts that are either numeric, in the range 0 through 9, or alphabetic. When you enter a user interrupt on a working connection, CDCNET synchronizes the connection and supplies the associated numeric or alphabetic character to the application as the reason code. This reason code is sent to the application as expedited data, and is therefore not subject to the flow control constraints of normal data (in other words, it is not queued behind other data in the input data buffers). It is the responsibility of the application to respond to the user interrupt.

If the interrupt character is a numeric in the range 1 through 9, CDCNET synchronizes the connection and discards any input and output that might currently be held in network queues for the connection. If the interrupt character is 0, CDCNET discards only the input. If the interrupt character is alphabetic, CDCNET does not discard input or output. These actions are summarized in the following table:

Interrupt Character	Action
1 through 9	The connection is synchronized and both input and output are discarded. The interrupt character is forwarded to the application as the reason code for the synchronization.
0	The connection is synchronized and only input is discarded. A reason code of 0 is forwarded to the application.
Alphabetic	The connection is synchronized, but neither input nor output is discarded. The interrupt character is forwarded to the application as the reason code for the synchronization.

If an additional user interrupt is entered before the network receives a response from the application to a user interrupt already being processed, the additional interrupt is ignored. User interrupts entered on the $NET connection are also ignored. In either case, CDCNET displays the following message:

```
User interrupt ignored.
```

You can enter user interrupts while in normal or transparent input editing mode. However, CDCNET recognizes a different type of user interrupt for each mode. You can enter a user interrupt by:

- Using the network command character
- Using the BREAK key or an attention character

In some cases, you can enter a numeric user interrupt *either* by using the network command character *or* by using the BREAK key or attention character. However, in situations when the network command character is not recognized by the network, you can *only* enter user interrupts with the BREAK key or attention character. The following sections describe how to enter user interrupts in each situation.

Entering User Interrupts with the Network Command Character

In normal input editing mode, a user interrupt can be entered using the network command character followed by a single alphabetic or numeric character. For example, the following entry causes the connection to be synchronized and a user interrupt 2 to be sent to the application:

```
%2
```

If you enter the network command character followed by a single character that is neither alphabetic nor numeric (for example, %$), CDCNET interprets the entry as an invalid user interrupt and issues the following message:

```
Invalid user interrupt.
```

User interrupts can also be entered using the attention character or BREAK key, as described next. These alternative methods might be necessary under some conditions, such as when your service connection is operating in transparent input editing mode.

Entering User Interrupts with the BREAK Key or Attention Character

In transparent input editing mode, you can only enter a user interrupt with the BREAK key or attention character. This is because during transparent mode the network does not sample your input for the network command character.

The BREAK_KEY_ACTION connection attribute defines the numeric interrupt that is processed when the BREAK key is used, if your terminal has a BREAK key.

If your terminal does not have a BREAK key, a user interrupt can be entered with the attention character. The character that acts as the attention character is defined by the ATTENTION_CHARACTER *terminal* attribute, described in chapter 3. The interrupt that is processed when you enter the attention character is a numeric in the range 0 through 9 defined by the ATTENTION_CHARACTER_ACTION *connection* attribute, described in chapter 4.

For example, suppose you have used the following command to set your BREAK_KEY_ACTION connection attribute to 2:

```
%CHACA BKA=2
```

Thereafter, if you press the BREAK key you execute the equivalent of a level 2 user interrupt. That is, the following two actions have the same effect:

- Entering %2 (in nontransparent input editing mode)
- Pressing the BREAK key (in transparent input editing mode)

Special Sequences

As described previously, when you press the BREAK key or enter the attention character, CDCNET synchronizes your working connection and sends the corresponding reason code to the connected service for it to act upon. At this point, CDCNET is poised to accept one of three special character sequences from the terminal. That is, the asynchronous TIP always tests the characters following a BREAK key or attention character for one of the following three special sequences:

- Secure access sequence
- Transparent input editing mode escape sequence
- Escape to $NET connection sequence

These special character sequences, which cannot be altered, alert CDCNET that you want to do one of the following special tasks, as described in the following sections.

- Delete all existing connections.
- Terminate transparent mode.
- Escape to the $NET connection while in transparent mode.

Delete All Existing Connections

When you want to delete all existing service connections, use the secure access sequence. The secure access sequence provides a trusted path to the $NET connection. At a site where network security is a concern, this sequence can be used to guarantee that your connection is to CDCNET, avoiding the scenario in which you provide your username and password to an application that merely imitates CDCNET and/or the host validation interface.

Delete all existing connections as follows:

1. Press the BREAK key or enter the attention character.
2. While holding down the control key (CTRL) press A, and then T.
3. Press NEXT (or return).

CDCNET deletes all existing service connections, resets all terminal and connection attributes to the CDCNET defaults, and issues its banner to your terminal, including the prompt:

```
You may enter CDCNET commands.
```

At this point, you are using the $NET connection and can enter any CDCNET commands that are valid on this connection.

Terminate Transparent Editing Mode

When your connection is in transparent input editing mode, CDCNET does not sample your input for the network command character. If you need to suspend your session with an application that is running in transparent mode in order to enter a CDCNET command, you can terminate transparent editing mode with the transparent mode escape sequence. Besides forwarding the break condition to the connected application, this sequence causes CDCNET to change your INPUT_EDITING_MODE connection attribute from TRANSPARENT to NORMAL. It is then possible to execute a CDCNET command that is prefixed with the network command character.

Terminate the transparent editing mode as follows:

1. Press the BREAK key or enter the attention character.
2. Hold down the control key (CTRL) and press X.
3. Press NEXT (or return).

Following this sequence, you could then enter a CDCNET command, which CDCNET would recognize. For example, if you are engaged in a session with an application that is running in transparent mode and you need to change a connection attribute, you could take the following steps:

1. Enter the transparent mode escape sequence.
2. Enter the CHANGE_CONNECTION_ATTRIBUTE command. For example,

   ```
   %CHACA TML=135
   ```

3. Return to transparent input editing mode with the following CHANGE_CONNECTION_ATTRIBUTE command:

   ```
   %CHACA IEM=T
   ```

Escape to the $NET Connection While in Transparent Mode

If you are using the Asynchronous or X.25 protocol and need to escape to the $NET connection, you can do so with the escape to $NET connection sequence. Besides forwarding the break condition to the connected application, this sequence causes CDCNET to change your working connection to the $NET connection without deleting your current working connection or any other existing connections.

If this sequence is used while the BREAK_KEY_ACTION connection attribute is set to 0, it is functionally equivalent to executing a %CHAWC command without parameters.

Escape to the $NET connection with the following special break sequence:

1. Press the BREAK key or enter the attention character.

2. Hold down the control key (CTRL) and press C.

CDCNET responds by issuing its banner message, and the prompt:

```
You may enter CDCNET commands
```

You can return to the suspended connection with a CHANGE_WORKING_CONNECTION command.

Interactive Terminal Passthrough 7

Interactive Terminal Passthrough 7

Interactive terminal passthrough connects two asynchronous users for the purpose of exchanging transparent data. The feature is provided by CDCNET's passthrough service.

Passthrough permits connections between asynchronous end-users, including the following:

- Interactive terminals and hosts foreign to CDCNET
- Interactive terminals and smart dial-out modems
- Two interactive terminals in a conversational mode
- Two microcomputers engaged in a file transfer

The term *passthrough user* refers to any of the types mentioned above. However, the text in this chapter is written for the interactive terminal user and microcomputer user.

All passthrough users fall into one of two categories:

User Type	Role Description
Passthrough Server	Offers services by defining a passthrough title to which passthrough clients can connect.
Passthrough Client	Connects to the services offered by a passthrough server.

This chapter describes how to become a passthrough server or passthrough client, and how to manage a passthrough connection. It also identifies the connection and terminal attributes that are important to passthrough connections, and describes the DEFINE_PASSTHROUGH_TITLE command.

Refer to chapter 9 for notes and cautions about using the interactive terminal passthrough service.

Becoming a Passthrough Server

To become a passthrough server, you must register a title with the passthrough service. This is a two-step process:

1. Create a connection to the passthrough service.

2. Define a passthrough title.

Creating a Connection to the Passthrough Service

Use the CREATE_CONNECTION (CREC) command described in this manual to establish a connection to the passthrough service. The service name of the passthrough service is site-configurable, with a default value of PASSTHROUGH.

For example, the following command creates a connection to the passthrough service if the site uses the default passthrough service name.

```
CREC PASSTHROUGH
```

The following message indicates that a connection to the passthrough service was made successfully:

```
Connection to Passthrough Service complete.
```

NOTE

The site must configure the passthrough service into the DI to which the intended server is connected. If the passthrough service is *not* installed when the server attempts to connect to the passthrough service, the CREATE_CONNECTION command fails and issues the "Cannot locate service" message.

Defining a Passthrough Title

After connecting to passthrough with CREC, the intended server must next register a server title with the passthrough service. This is done with the DEFINE_PASSTHROUGH_TITLE (DEFPT) command, described later in this chapter.

If the connection to CDCNET passthrough was made by a terminal user command (that is, CREC PASSTHROUGH_SERVICE or DO TUP_NAME), the title that is registered is prefixed by the two-character string, T_. This signifies that the title was registered by an interactive terminal user, as opposed to site configuration.

If the connection to CDCNET passthrough was made by a TUP that executed automatically when the user connected to the network, the title is registered as it appears on the DEFPT TITLE parameter. The absence of the T_ prefix indicates that the title was registered as the result of site configuration.

The first input after the passthrough connection is created must be a DEFPT command. If it is not, the intended server is prompted with the following message:

```
Passthrough input discarded.
Please register titles.
```

NOTE

Multiple DEFPT commands with the same title are allowed on the same network. This allows passthrough users to access one of many services providing the same function; for example, dial-out modems.

Becoming a Passthrough Client

After one passthrough user becomes a registered server, another user becomes a passthrough client by creating a connection to the server's registered title. This is done using the CREATE_CONNECTION (CREC) command. Specify a SERVICE_NAME value equal to the title registered by the passthrough server.

You are notified if the connection is successful. Message text differs between connections to servers that registered through site configuration and those that registered interactively. Successful connection to a site-defined service brings the following response:

```
Connection to Site Passthrough service service_name complete.
System = system_name, Line = line_name.
```

Successful connection to a user-defined service brings this response:

```
Connection to User Passthrough service service_name complete.
System = system_name, Line =line_name.
```

In either case, the system and line fields indicate the DI system and line names associated with the server.

Managing Passthrough Connections

Each passthrough server or passthrough client has options for managing its passthrough connection(s). These options are listed below and described afterward:

- Putting a passthrough connection on hold
- Temporarily entering normal input editing mode
- Letting a passthrough connection time out
- Deleting a passthrough connection
- Using the attention character or BREAK key
- Controlling passthrough service messages

Putting a Passthrough Connection On Hold

A working passthrough connection can be put on hold if the passthrough user (client or server) enters the escape to $NET connection sequence, as described in chapter 6.

After entering this special break sequence, you can enter any CDCNET terminal user commands. For example, you can create another connection with the CREATE_CONNECTION (CREC) command, or return to existing connections (including the passthrough connection) using the CHANGE_WORKING_CONNECTION (CHAWC) command; all CDCNET terminal user commands are described in this manual.

Entering Normal Input Editing Mode

The passthrough service only permits communication between passthrough client and server in the transparent input editing mode. In transparent mode you cannot execute CDCNET terminal user commands. However, if you enter the transparent mode escape sequence on this connection, you are temporarily placed in normal mode, from which you can enter terminal user commands.

The transparent mode escape sequence is as follows:

1. Press the BREAK key (or ATTENTION_CHARACTER).

2. Hold down the CONTROL (CTRL) key and press X.

You cannot communicate through passthrough using normal mode, but you *can* enter terminal user commands that are prefixed with the network control character. For example, %DISPLAY_COMMAND_LIST, %DISPLAY_CONNECTION_ATTRIBUTES, or %CHANGE_WORKING_CONNECTION.

The next time you enter something on this connection that is not prefixed with the network control character, the passthrough service resets the connection to transparent input editing mode and discards the input, issuing the following message:

```
Passthrough input discarded.
Passthrough connection attributes reset.
```

The preferred method of resuming transparent mode communication on the passthrough connection is to enter a single end-of-line character. The following message is issued:

```
Passthrough connection attributes reset.
```

This message indicates that the passthrough service has reset your input editing mode to transparent, and that passthrough communication can resume.

Letting a Passthrough Connection Time Out

A passthrough connection is considered idle if there is no data being passed in either direction. Some passthrough servers use an inactivity timer to disconnect idle connections (refer to the DEFINE_PASSTHROUGH_TITLE command described later in this chapter). When this timer approaches a full count, the following message, followed by the ASCII BEL character, is issued to the passthrough user:

```
Passthrough connection connection_name timeout in 30 seconds.
```

If the passthrough connection remains idle for an additional 30 seconds, the following message is issued to the terminal user, and the passthrough connection is disconnected:

```
Passthrough connection connection_name timeout.
```

Deleting a Passthrough Connection

Passthrough users (either servers or clients) can delete a passthrough connection in one of the following ways:

- By entering a service command to delete the connection, or
- By entering a $NET escape sequence (refer to the prior discussion about putting a passthrough connection on hold). The passthrough connection is then deleted using the DELETE_CONNECTION (DELC) command, described in this manual.

When a *client* deletes the connection, the following message is issued to the *server*:

```
Passthrough client connection disconnected.
```

When a *server* deletes the connection, the following message is issued to the *client*, and the client's connection is disconnected:

```
Passthrough server connection disconnected.
```

These messages are also issued to clients and servers when the network operator discontinues the passthrough service.

Using the Attention Character or BREAK Key

When you enter the attention character or press the BREAK key, the passthrough service receives a break indication with the BREAK_KEY_ACTION value equal to 0. To send a break across the passthrough connection to the other passthrough user, you must issue the equivalent of two break indications (press the BREAK key twice or enter two attention characters). A break indication that is not immediately followed by another break indication is discarded.

NOTE

Because the BREAK key and ATTENTION_CHARACTER actions use level zero user interrupts, some terminal user input may be lost.

Controlling Passthrough Service Messages

Passthrough service messages are conditionally issued to users based on the value of the RESPONSE_ACTION (RA) terminal attribute. When you first connect to the passthrough service, the service checks the value of the RA attribute. If RA=SEND, then all passthrough-generated messages are issued directly to your connection. If RA=DISCARD, then these messages are not issued.

NOTE

Passthrough only checks the RA attribute value when the connection is first made. Changing the value of RA after the connection is made has no affect on whether passthrough service messages are forwarded to you. Thus, the appropriate RA value should be selected before you connect to passthrough.

Passthrough Attributes

Both the client and server connections are put in transparent input editing mode when a passthrough client connects to a passthrough server. This gives the server and/or client complete control of the input editing and output formatting.

Connection Attributes

The following passthrough connection attribute values are set in a passthrough connection, and *cannot* be changed.

```
INPUT_EDITING_MODE     = TRANSPARENT
INPUT_OUTPUT_MODE      = FULLDUPLEX
BREAK_KEY_ACTION       = 0
ATTENTION_KEY_ACTION = 0
```

INPUT_OUTPUT_MODE is set to full duplex so that the client and server can emulate the input/output mode of their choice.

Both the BREAK_KEY_ACTION and ATTENTION_CHARACTER_ACTION are set to zero so that only input is synchronized when passing the break condition through the network. If you try to change either of these values they are reset immediately to zero by the passthrough service.

The default values for passthrough's data-forwarding connection attributes are as follows:

```
INPUT_BLOCK_SIZE                      = 2000
TRANSPARENT_CHARACTER_MODE            = NONE
TRANSPARENT_FORWARDING_CHARACTERS = CR  LF  8A(16) 8D(16)
TRANSPARENT_LENGTH_MODE               = FORWARD
TRANSPARENT_MESSAGE_LENGTH            = line_speed divided by 40
TRANSPARENT_TIMEOUT_INTERVAL          = 2
TRANSPARENT_TIMEOUT_MODE              = FORWARD
```

NOTE

These connection attribute settings are appropriate for most passthrough connections, and result in the best network performance. Although you may change connection attributes, Control Data does not recommend it.

Terminal Attributes

The ATTENTION_CHARACTER terminal attribute should be set to NUL for passthrough connections. Otherwise, you cannot transmit the attention character to the destination because it is processed the same as the BREAK key.

Using Flow Control

Flow control on passthrough connections is highly recommended. Use the CHANGE_TERMINAL_ATTRIBUTES (CHATA) command to change character flow control. The following are guidelines for helping you choose the correct flow control:

- For dedicated lines, use EIA flow control with RTS/CTS signals, if your cabling and your device support it. Use of EIA flow control must be coordinated with your site administrator.

- Configurations that do not support EIA flow control (certain direct connections to hosts and microcomputers, for example), should use character flow control.

- Character flow control should also be used for connections involving a dial-out modem.

- For binary file transfers, use character flow control, but only if the binary data does not include embedded X-ON/X-OFF characters; these characters in binary data can cause the file transfer to suspend indefinitely. (For example, because XMODEM allows X-ON/X-OFF characters to be part of data, do not use character flow control.)

It may be possible to do without flow control for low-speed, low-volume data transfers, but only if the file transfer protocol can recover from lost or partially lost file transfer data units.

DEFINE_PASSTHROUGH_TITLE (DEFPT)

Purpose

Registers the title(s) for a passthrough server, enabling another person or service to connect to your server for the exchange of data.

The DEFINE_PASSTHROUGH_TITLE command should not be prefixed with the network command character. DEFPT is a command that is acted upon by the passthrough service, not by $NET.

Format

DEFINE_PASSTHROUGH_TITLE
  TITLE=name
  INACTIVITY_TIMER=integer or *keyword*

Parameters

TITLE (T)

Specifies the title(s) to be registered with the passthrough service. These are the names that a passthrough client uses on the CREATE_CONNECTION command. If you specify more than one name, enclose the names in parentheses and separate them with spaces.

INACTIVITY_TIMER (IT)

Specifies the maximum time, in seconds, that a passthrough connection can remain idle. If no data is transferred in either direction on the connection for the specified length of time, the passthrough connection is disconnected. Allowed values are any integer from 120 through 14400, or the keyword INFINITE. If you specify INFINITE, the network does not disconnect the connection because of inactivity.

If you do not specify a value for this parameter, the network uses the value the site administrator specified when the passthrough service was defined.

Remarks

If the connection to CDCNET passthrough was made by a terminal user command (that is, CREC passthrough_service or DO tup_name), the title that is registered is prefixed by the two-character string, T_. This signifies that the title was registered by an interactive terminal user, as opposed to site configuration.

If the connection to CDCNET passthrough was made by a TUP that executed automatically when the user connected to the network, the title is registered as it appears on the DEFPT TITLE parameter. The absence of the T_ prefix indicates that the title was registered as the result of site configuration.

Multiple DEFPT commands with the same title are allowed on the same network. This allows passthrough users to access one of many services providing the same function; for example, dial-out modems.

Examples

Suppose you connect a microcomputer to the network that is able to provide services to other microcomputers (such as electronic mail management). You make these services available on the network in the following steps:

1. Create a connection to the passthrough service (assuming your site uses the default passthrough service name, PASSTHROUGH):

   ```
   CREC PASSTHROUGH
   ```

2. Register as a passthrough server with the following command:

   ```
   DEFPT T=MAILPC IT=INFINITE
   ```

Subsequently, other microcomputers on the network can connect to your service with the following command:

```
%CREC T_MAILPC
```

Responses

If the DEFPT command is successful, the following response is returned:

```
Passthrough service titles registered.
```

You can now receive connections from passthrough clients.

If, for any reason, the titles cannot be registered in the CDCNET directory, the following message is returned:

```
Passthrough service titles not registered.
```

In this case, you should enter the DEFPT command again.

X.25 Interactive Terminal Gateway 8

X.25 Interactive Terminal Gateway 8

The X.25 interactive terminal gateway allows asynchronous users to access an X.25 service or a remote DTE on an X.25 public data network (PDN) with CDCNET. As seen in figure 8-1, users can be physically connected anywhere on the network and have access to the gateway via CDCNET.

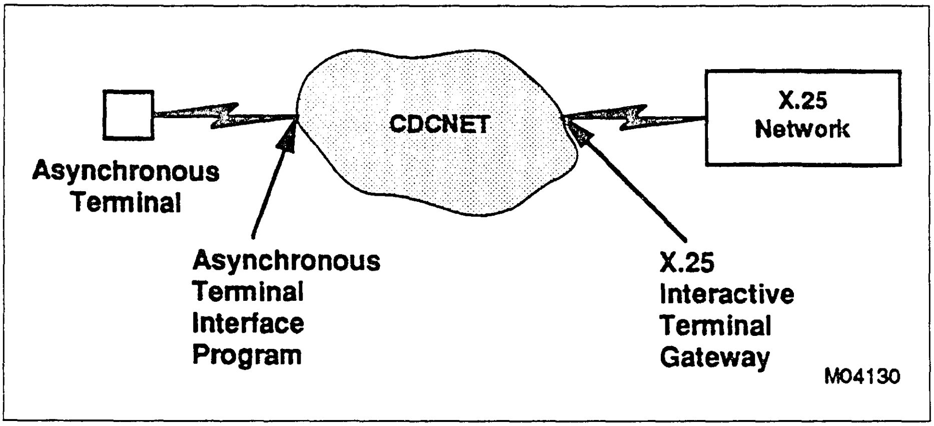


Figure 8-1. X.25 Interactive Gateway Environment

X.25 Interactive Terminal Gateway Modes

The gateway has two operating modes; both are discussed in this chapter:

- Gateway command mode
- Data transfer mode

Gateway Command Mode

Gateway command mode allows you to communicate with the X.25 terminal gateway. Once you have created a connection to the X.25 terminal gateway, you can direct it to perform one of several terminal gateway commands, to modify one of several parameters affecting X.25 communications or to connect to the remote DTE. You communicate with the gateway via several commands, which you enter following a gateway command prompt. The commands available to you in gateway command mode are discussed later in this chapter. The gateway command prompt is:

```
x25/
```

Gateway Command Mode Connection Attributes

Once you have created a connection to the X.25 terminal gateway, your connection attributes are set by CDCNET as shown in figure 8-2.

```
Attention_Character_Action          : 2
Attention_Character_Enable          : ON
Break_Key_Action                    : 0
Echo_Enable                         : ON
Input_Block_Size                    : 160
Input_Editing_Mode                  : NORMAL
Input_Flow_Control_Enable           : ON
Input_Output_Mode                   : UNSOLICITED
Output_Flow_Control_Enable          : ON
Parity_Enable                       : ON
Partial_Character_Forwarding        : OFF
Store_Backspace_Character           : OFF
Store_Nuls_Dels                     : OFF
Transparent_Character_Mode          : TERMINATE
Transparent_Forward_Character       : CR 8D(16)
Transparent_Length_Mode             : NONE
Transparent_Message_Length          : 255
Transparent_Protocol_Mode           : NONE
Transparent_Terminate_Character     : CR 8D(16)
Transparent_Timeout_Interval        : 0
Transparent_Timeout_Mode            : NONE
```

Figure 8-2. Gateway Command Mode Connection Attribute Defaults

NOTE

These connection attributes are appropriate for most X.25 gateway connections, and result in the best network performance. Although you may change connection attributes, Control Data does not recommend it.

Once you are connected to the gateway in command mode, you may change any connection attribute using the CHACA terminal interface command as described in chapter 4. CDCNET recognizes the changed attribute value when the X.25 session is established with the X.25 service.

Data Transfer Mode

Data transfer mode allows you to communicate interactively with a remote DTE. The prompt you receive once your terminal is in data transfer mode depends upon the output from your remote DTE.

Data Transfer Mode Connection Attributes

Once the remote DTE accepts the incoming call from your terminal (sent while your terminal is in gateway command mode), the gateway enters data transfer mode. The gateway then sets the connection attributes for the remote DTE connection. If you activated a TUP when you created the connection to the gateway, your connection attributes may differ from the default connection attributes (figure 8-3) recognized by CDCNET after establishment of data transfer mode:

```
Attention_Character_Action          : 0
Attention_Character_Enable          : OFF
Break_Key_Action                    : 0
Echo_Enable                         : ON
Input_Block_Size                    : set to the X.25 packet size
Input_Editing_Mode                  : TRANSPARENT
Input_Flow_Control_Enable           : ON
Input_Output_Mode                   : FULLDUPLEX
Output_Flow_Control_Enable          : ON
Parity_Enable                       : ON
Partial_Character_Forwarding        : ON
Store_Backspace_Character           : ON
Store_Nuls_Dels                     : ON
Transparent_Character_Mode          : FORWARD
Transparent_Forward_Character       : CR 8D(16) DLE 90(16)
Transparent_Length_Mode             : FORWARD_EXACT
Transparent_Message_Length          : set to the X.25 packet size
Transparent_Protocol_Mode           : NONE
Transparent_Terminate_Character     : NUL
Transparent_Timeout_Interval        : 2
Transparent_Timeout_Mode            : FORWARD
```

Figure 8-3. Data Transfer Mode Connection Attribute Defaults

NOTE

These connection attribute settings are appropriate for most X.25 connections. If changes to these default settings are required, they can be made using the %CHACA command.

Once you are operating in data transfer mode, you may change connection attributes, but only to correct a well-defined problem. This is discussed in Changing Gateway Connection Attributes later in this chapter.

X.3 Packet Assembler/Disassembler (PAD) Parameters

When connected to the X.25 service, certain X.3 packet assembler/disassembler (PAD) parameter values set by the remote X.25 service are mapped to a corresponding connection or terminal attributes supporting your terminal connection. If your terminal supports the X.25 asynchronous protocol, the X.3 PAD parameters are mapped to certain attributes as shown in table 8-1. If your terminal does not support the X.25 asynchronous protocol, the X.3 PAD parameters are mapped to certain terminal attributes as shown in table 8-2.

Table 8-1. Pad Parameter Mapping Supporting the X.25 Asynchronous Protocol

PAD Parameter Number	PAD Parameter Name	Maps to Attribute
2	Echo	Echo_Enable
3	Forwarding Characters	Transparent_Forward Character
4	Idle Timer Delay	Transparent_Timeout_Interval Transparent_Timeout_Mode
5	Ancillary Device Control	Input_Flow_Control_Enable
12	Flow Control	Output_Flow_Control_Enable
21	Parity Treatment	Parity_Enable

Table 8-2. PAD Parameter Mapping Supporting Non-X.25 Asynchronous Protocols

PAD Parameter Number	PAD Parameter Name	Maps to Attribute
2	Echo	Echoplex
3	Forwading Characters	No attribute changes
4	Idle Timer Delay	Transparent_Timeout_Interval Transparent_Timeout_Mode
5	Ancillary Device Control	Character_Flow_Control
12	Flow Control	Character_Flow_Control

Connecting to the X.25 Interactive Terminal Gateway

After accessing CDCNET via an asynchronous terminal interface program (TIP), you can connect to an X.25 interactive terminal gateway with the CREATE_CONNECTION terminal user command. The SERVICE_NAME parameter of the command must specify the title of the gateway to which you want to connect. Your site administrator knows the gateway titles to which you have access.

Following successful connection to the specified gateway, you receive the CDCNET connection message and the gateway command prompt:

Connection *connection_name* created
x25/

You can now create an X.25 connection (with the CREATE_X25_CONNECTION terminal gateway command) to the X.25 service to which you want to communicate, or enter any of the supported X.28 commands. These commands are all described later in this chapter.

Refer to figure 8-4 when reading the next paragraphs describing how you connect to an X.25 service and a specific DTE address on that service.

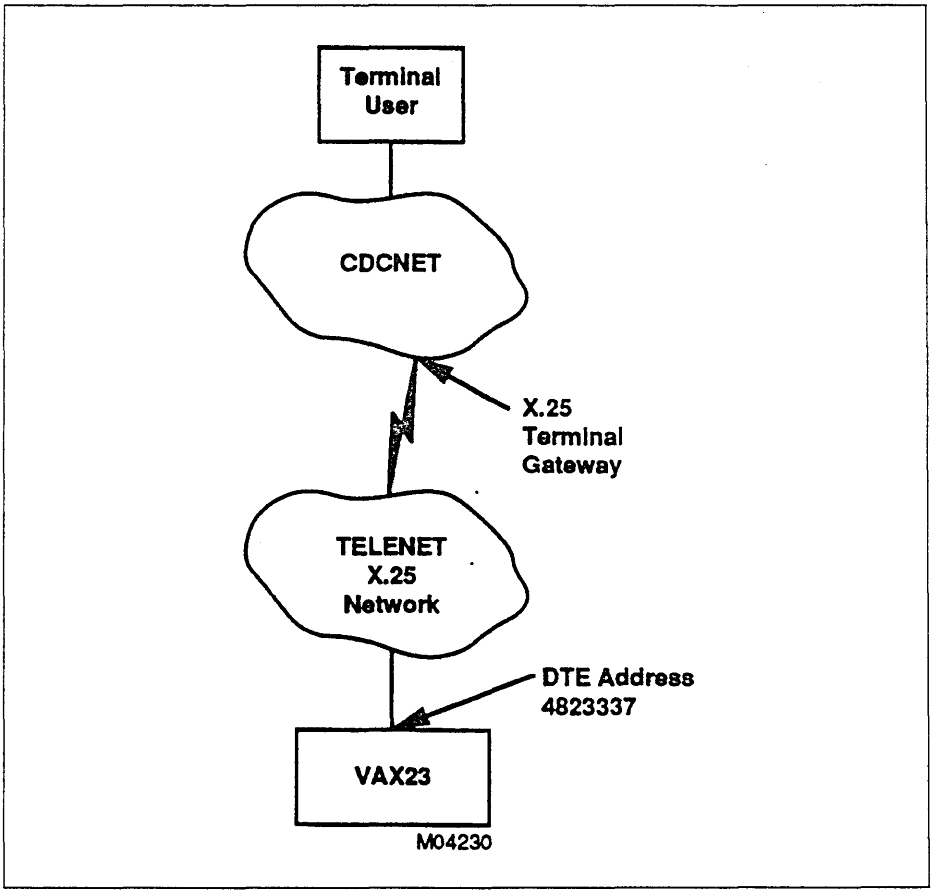


Figure 8-4. Connecting to a DTE on an X.25 Network.

Connecting to the X.25 Service

You can connect to the X.25 service in one of two ways:

- Create the connection in two steps with the CREATE_CONNECTION terminal user command and the CREATE_X25_CONNECTION gateway command.

- Create the connection in one step with the CREATE_CONNECTION terminal user command.

Connecting to the X.25 Service in Two Steps

You can connect to the X.25 service in two steps as follows:

1. Create a connection to the X.25 interactive terminal gateway available to you using the CREATE_CONNECTION terminal user command. Specify the title of the gateway to which you want to connect on the SERVICE_NAME parameter.

   ```
   CREC SN=TELENET
   ```

   CDCNET returns the CDCNET connection message and the gateway command mode prompt:

   ```
   Connection $A created
   x25/
   ```

2. Create a connection to the remote DTE with which you want to communicate using the CREATE_X25_CONNECTION gateway command mode command. Specify the destination as either a valid remote DTE address or a valid X.25 service name. The following command requests a connection to a specific service name.

   ```
   CREXC SN=4823337
   ```

Following the successful execution of the CREXC command, CDCNET returns the banner and login prompt of the X.25 service to which you are now connected.

Refer to the description of the CREATE_X25_CONNECTION command later in this chapter for more information.

Connecting to the X.25 Service in One Step

You can connect to the X.25 service in a single step using the CREATE_ CONNECTION terminal user command to connect to the X.25 gateway and to establish the X.25 connection with the remote DTE.

Enter the CREATE_CONNECTION command, specifying values for the SERVICE_NAME parameter and the SERVICE_DATA parameter.

```
CREC SN=TELENET SD=4823337
```

Following the successful execution of the CREC command, CDCNET returns the banner and login prompt of the remote DTE to which you are now connected.

Managing the X.25 Service Connection

Once established, you can manage the X.25 connection from your terminal in the following ways:

- Changing gateway connection attributes
- Switching between gateway modes
- Deleting a gateway connection

Changing Gateway Connection Attributes

You can change gateway connection attributes when you are in either gateway command mode or in data transfer mode. Changing connection attributes from either mode involves some risk.

CAUTION

Changing attributes before or during an X.25 session may prevent your terminal from transmitting to the remote DTE, or prevent the remote DTE from responding to your terminal.

Should you need to change a connection attribute while in gateway command mode, you can change a connection attribute via the CHANGE_CONNECTION_ATTRIBUTES terminal user command. CDCNET recognizes the new value specified by the CHACA command when you establish the X.25 session with the remote DTE. This occurs when the gateway enters data transfer mode.

Refer to chapter 2 for a description of the CHACA command.

Switching Between Gateway Modes

Once you are in data transfer mode, you may need to interrupt the remote session, either temporarily or permanently switching to gateway command mode.

Switching to Gateway Command Mode from Data Transfer Mode

You can switch to gateway command mode from data transfer mode using one of two PAD escape sequences or a CDCNET interrupt sequence via the BREAK key.

- If CCITT X.3 parameter number 1 (Pad Recall Character) value is 1 (Pad Recall With DLE), enter CTRL (P), and press RETURN. Your screen displays the following gateway command prompt:

  ```
  x25/
  ```

- If CCITT X.3 parameter number 7 (Break Action) value is 8 (Escape from Data Transfer), press the BREAK key twice. This restores the connection to the gateway command mode. This double BREAK key processing depends entirely on the value of the X.3 parameter 7. You cannot pass a BREAK signal to the remote DTE when the specified action is to escape from the data transfer mode. Your screen displays the following gateway command prompt:

  ```
  x25/
  ```

- If you want to interrupt data transfer mode temporarily by resetting the input editing mode attribute to NORMAL, enter the escape sequence: BREAK CTRL (X). Input editing mode attribute is reset to NORMAL, but the terminal remains in data transfer mode. However, as soon as the gateway receives character input in normal mode, the gateway interprets it as a command and the gateway switches from data transfer mode to gateway command mode. Your screen then displays the following gateway command prompt:

  ```
  x25/
  ```

To determine the value of CCITT X.3 parameter numbers 1 (Pad Recall Character) and 7 (Break Action) use the gateway command mode command X.28 command PAR?. This command is discussed later in this chapter.

Switching to Data Transfer Mode from Gateway Command Mode

You can switch back to data transfer mode from gateway command mode by entering a carriage return in response to the X25/ prompt.

```
x25/<carriage return>
```

Deleting a Gateway Connection

You can delete an X.25 gateway connection from either gateway command mode or data transfer mode, as follows:

- To delete a gateway connection while in gateway command mode, enter the QUIT gateway command following the gateway command mode prompt:

  ```
  x25/QUIT
  ```

- To delete a gateway connection while in data transfer mode, you must first interrupt data transfer mode using either an X.3 PAD escape sequence described in the preceding section. Enter the CLR X.28 command following the gateway command mode prompt.

  ```
  x25/CLR
  CLR CONF
  ```

  The connection to the remote DTE is now deleted. You can now delete the connection in gateway command mode with the QUIT command.

X.25 Interactive Terminal Gateway Commands

There are two kinds of X.25 interactive terminal gateway commands. Terminal gateway commands allow you to control several aspects of gateway operation, and X.28 commands affect certain X.28 parameters affecting X.25 communication.

X.25 terminal gateway commands are discussed in the following paragraphs. X.28 commands are discussed later in this chapter.

Terminal Gateway Commands

Terminal gateway commands allow you to create the connection with the remote DTE on an X.25 PDN, to display command and service information available to you within the gateway, and to terminate the connection with the gateway.

The terminal gateway commands include the following commands that follow the SCL syntax conventions:

CREATE_X25_CONNECTION

DISPLAY_COMMAND_INFORMATION

DISPLAY_COMMAND_LIST_ENTRY

DISPLAY_X25_SERVICES

QUIT

CREATE_X25_CONNECTION (CREXC)

Purpose

Creates a connection to a remote DTE on a public data network. You can specify either a site-defined name or the decimal DTE address of the remote DTE to which you want to connect. Before specifying the remote DTE to which you want to connect, you may want to check with your site administrator to learn which of the following addressing options are available to you:

- Specify a DTE address that falls within a specified range of DTE addresses. For example, if a range of 3110408* is configured, only DTE addresses beginning with 3110408 are accepted.
- Specify a service name, which is associated with one or more specific remote DTEs. For example, a particular service name is mapped to one or more specific remote DTEs, depending upon your configuration.
- Specify a complete remote DTE address.

Format

CREATE_X_25_CONNECTION
SERVICE_NAME = name, or string 1..15

Parameters

SERVICE_NAME (SN, REMOTE_DTE_ADDRESS, RDA)

This parameter identifies the remote DTE to which you want to make the X.25 connection. The following parameter values are valid:

- SN – A parameter entered as a service name must match a site-configured title associated with a complete remote DTE address.
- REMOTE_DTE_ADDRESS or RDA – A parameter entered as a string can contain only the characters 0 through 9, and must satisfy the site-configured outcall blocks, if the gateway requires it.

Responses CDCNET issues the following responses after receiving an incorrect CREXC command:

--ERROR--Unauthorized remote DTE address specified.
--ERROR-- A remote_dte_address may include only digits 0 through 9.
--ERROR-- An X.25 connection currently exists.
--ERROR-- Unable to locate service *service_name*.
--ERROR-- Unauthorized remote_dte_address specified.
--ERROR-- Too many parameters given.
--ERROR-- Parameter **SERVICE_NAME** is required but was omitted.
--ERROR-- Connection rejected. No memory is available.
--ERROR-- Connection rejected. All X.25 logical channels are currently busy.
--ERROR-- Connection rejected. The network is currently congested.
--ERROR-- Connection rejected. The local dte address is currently inoperative.
--ERROR-- Connection broken. The X.25 interface has been stopped by the network operator.
--ERROR-- Connection broken. The X.25 link is now inoperative.
--ERROR-- Connection broken. The X.25 link is currently experiencing problems.
--ERROR-- Connection broken. The remote DTE has cleared the connection. Clearing cause code=XX and diagnostic code=YY.
--ERROR-- Connection broken. The local DTE has cleared the connection. Clearing cause code=XX and diagnostic code=YY.
--ERROR-- Connection broken. The X.25 link has temporarily gone down.
--ERROR-- Connection to X.25 Terminal Gateway broken. The network operator has stopped the X.25 Terminal Gateway.

Examples The following examples assumes the following outcall blocks have been configured: name1=3110408*, name2=1311061200123.

The following command creates a connection to a remote DTE specifying a service name.

```
x25/CREXC SN=name2
```

Your terminal displays the banner and login prompt from the remote DTE.

The following command creates a connection to a remote DTE by specifying the complete remote DTE address.

```
x25/CREXC RDA=311040800123
```

Your terminal displays the banner and login prompt from the remote DTE.

The following command attempts to create a connection to a remote DTE by specifying the complete remote DTE address.

```
x25/CREXC RDA=62600123
```

Your terminal displays the following message:

```
--ERROR--Unauthorized remote DTE address specified.
```

DISPLAY_COMMAND_INFORMATION (DISCI)

Purpose

Displays the parameters and parameter syntax information for a specified X.25 terminal gateway command and the associated X.28 commands. The specified command must be one of the available X.25 gateway or X.28 commands.

Format

DISPLAY_COMMAND_INFORMATION
COMMAND = name of command

Parameters

COMMAND (C)

Specifies the command for which the parameters are to be displayed. You must provide either the full command name or the command abbreviation. The specified command must be one of the following X.25 terminal gateway or X.28 commands:

CREATE_X25_CONNECTION

DISPLAY_COMMAND_INFORMATION

DISPLAY_COMMAND_LIST_ENTRY

DISPLAY_X25_SERVICES

QUIT

CLR

INT

PAR?

SET

STAT

Responses

List of paramter names, parameter abbreviations, and parameter syntax for the specified command. (See example.)

--ERROR--*xxxx* is not a command.
--ERROR-- Parameter COMMAND is required but was omitted.
--ERROR-- Too many parameters given.

Examples

The following command requests a display of the parameters and syntax of the CREXC command.

```
x25/DISCI CREXC
```

Your terminal displays the following information:

```
service_name,sn,remote_dte_address,rda: string 1..15
```

or

```
$name = $required
```

The following command requests a display of the parameters and syntax of the X.28 command SET:

```
x25/DISCI SET
```

Your terminal displays the following information:

```
CCITT syntax - number1:value,number2:value ..
number22:value
```

DISPLAY_COMMAND_LIST_ENTRY (DISCLE)

Purpose Displays the commands available after you have established a connection to the X.25 terminal gateway and the gateway is in gateway command mode. CDCNET issues the gateway command mode prompt (x25/) to your terminal when you enter gateway command mode.

Format **DISPLAY_COMMAND_LIST_ENTRY**

Parameters None.

Responses List of commands available while you are in gateway command mode. (See example.)

--ERROR-- Too many parameters given.

Examples The following command requests a list of the commands available during the current X.25 terminal gateway session:

```
x25/DISCLE
```

Your terminal displays the following information:

```
X.25 Terminal Gateway Commands

create_x25_connection
display_command_information
display_command_list_entry
display_x25_services
quit

X.28 Commands

clr
int
par?
set
stat
```

DISPLAY_X25_SERVICES (DISXS)

Purpose Displays the list of site-configured service names with their associated remote DTE addresses. Note that the fully specified service names with no wildcard characters in their associated remote address can be used in place of the address on a CREC or CREXC command.

Format **DISPLAY_X25_SERVICES**

Parameters None.

Responses List of service names and their associated remote DTE addresses. (See example.)

--ERROR-- Too many parameters given.

Examples The following example lists the service names and associated remote DTE addresses:

```
x25/DISXS

Service Name            Remote DTE Address
NAME1                   33333333
NAME2                   22222222
NAME4                   44444444
NAME1                   11111111
n/a                     3110612*
n/a                     3110408*
n/a                     3110*
n/a                     *
```

The following example shows the response when your site does not require gateway outcall blocks, and access is unlimited; that is, you can request that an X.25 connection be made to any remote DTE:

```
x25/DISXS

Service Name              Remote DTE Address

Unlimited Access
```

QUIT (QUI)

Purpose Terminates your connection with the X.25 terminal gateway. The command also breaks the corresponding connection with the remote DTE, if one exists.

Format **QUIT**

Parameters None.

Responses You receive the following response if the X.25 connection successfully terminates:

```
You may enter CDCNET commands.
```

--ERROR-- Too many parameters given.

Examples The following example shows the use of the QUIT command.

```
x25/QUIT
```

CDCNET sends the following response to your terminal.

```
You may enter CDCNET commands.
```

X.28 Commands

X.28 commands allow you to change or request the status of X.3 PAD parameters affecting X.25 communications with a remote DTE.

The X.28 commands include the following commands that follow the CCITT Recommendation X.28 syntax conventions:

CLR

INT

PAR?

SET

STAT

CLR

Purpose Terminates your X.25 connection with the remote DTE.

Format CLR

Parameters None.

Responses You receive the following response if the X.25 connection with the remote DTE successfully terminates.

CLR CONF

--ERROR-- Too many parameters given.

Examples The following example shows the use of the CLR command.

```
x25/CLR
```

CDCNET sends the following response to your terminal.

```
CLR CONF
```

INT

Purpose Sends an interrupt packet to the remote DTE.

Format INT

Parameters None.

Responses You receive one of the following responses if the interrupt command was not successful.

INT ERR

--ERROR-- Too many parameters given.

Examples The following example shows the use of the INT command.

```
x25/INT
```

PAR?

Purpose Displays the current values of the X.3 parameter reference numbers.

Format **PAR?**

Parameters None.

Responses You receive a display of the current values of the X.3 reference numbers.

--ERROR-- Too many parameters given.

Examples The following example shows the use of the PAR? command.

```
x25/PAR?

PAR1:0,2:0,3:2,4:0,5:0,6:0,7:21,8:0,9:0,10:0,11:0,
12:0,13:0,14:0,15:0,16:0,,17:0,18:0,19:0,20:0,
21:0,22:0
```

SET

Purpose Changes the curent values of the specified X.3 parameter reference numbers. You can only change the reference numbers supported by the X.25 interactive terminal gateway.

Format SET
NUMBER:VALUE,NUMBER:VALUE
..NUMBER:VALUE

Parameters NUMBER

Specifies the X.3 reference number of the parameter to be changed.

VALUE

Specifies the value to which the X.3 reference number parameter is to be changed.

Responses Your terminal displays the gateway command prompt x25/ when the SET command completes successfully. (See example.)

PAR n:INV

PAR INV:N

SET ERR

Examples The following example shows the use of the SET command. The X.3 reference number parameter 1 value is changed to 1, and the X.3 reference number parameter 7 value is changed to 8.

```
x25/SET 1:1,7:8

x25/
```

STAT

Purpose Checks and displays the state of the X.25 connection to the remote DTE.

Format STAT

Parameters None.

Responses You receive the following response if no X.25 connection exists with a remote DTE.

FREE

You receive the following response if an X.25 connection currently exists with a remote DTE.

ENGAGED

You receive the following response if you have specified too many parameters.

--ERROR-- Too many parameters given.

Examples The following example shows the use of the STAT command.

```
x25/STAT

ENGAGED
```

Resolving Communications Problems 9

Resolving Communications Problems 9

This chapter provides some notes and cautions regarding network communications problems. Understanding these concepts can help you to resolve these problems.

- Network failures
- Automatic recognition difficulties
- Inadvertent flow control
- Accidental transparent input
- Passthrough service difficulties
- Service failures
- Cursor positioning delays

Network Failures

The network may not respond when you enter commands. This lack of response means the network equipment (device interface) you are using may not be functioning properly. Therefore, CDCNET cannot respond to your commands. Wait 5 to 10 minutes, and then try to access CDCNET again. If it still does not respond, call your site administrator.

If your line is defined to use automatic recognition and the network is not responding to your commands, you may be experiencing a problem with automatic recognition. See Automatic Recognition Difficulties, next in this chapter, for information on how to solve this problem.

If you are accessing CDCNET from a PDN, non-CDCNET LAN, or other multiplexing equipment, the problem may be in the other network or equipment rather than CDCNET. As part of your approach to solving the problem, consult the appropriate non-CDCNET documentation.

If you succeed in accessing the network after the failure, or you accessed your service via an alternate route (different device interface), you may discover that your service still lists your previous job as active. For example, you might check job status by entering a service command (such as DISPLAY_JOB_STATUS on NOS/VE).

You can try to recover this job. If you wait 5 to 10 minutes, the service should suspend the job due to inactivity and then you may be able to recover it.

Automatic Recognition Difficulties

Your terminal may encounter network communications difficulties during the automatic recognition process:

- When you are initially accessing the network and press the RETURN key (or equivalent) twice.
- When you are using the ACTIVATE_AUTO_RECOGNITION command. (See chapter 2 for a description.)

If your terminal does not respond to your input, perform the following steps:

1. Press the BREAK key.
2. Then press the RETURN key twice.

In response, you should receive some output from the network. If you do not, take the following steps:

1. Turn off the power to your terminal.
2. If your terminal has a switch for coded character set (also called character set, code set, etc.), be sure it is set correctly. It is usually ASCII.
3. If you are using a switched line, turn on the power to your terminal and follow your local procedure for connecting a terminal through a modem or data set to the computer.

   If your terminal is connected to a dedicated line, turn the power to your terminal on and wait 5 seconds.
4. Press the BREAK key again.
5. Then press the RETURN key twice.

Inadvertent Flow Control

Sometimes output stops and your terminal does not respond to your input. In these instances, your terminal may have asked the network to stop sending output again.

This happens when you are using the following terminal-attribute setting and the terminal sends an X-OFF character (DC3) to CDCNET.

CHARACTER_FLOW_CONTROL=ON

To reactivate the flow of data, press the CONTROL key and hold it down while you press Q. This control sequence forces the terminal to send the network an X-ON character (DC1), which directs CDCNET to send your terminal output again.

Accidental Transparent Input

If output stops and your terminal does not respond to your input, it might be because the INPUT_EDITING_MODE connection attribute is inadvertently set to TRANSPARENT. If you suspect this to be the case, enter the transparent input editing mode escape sequence, described in chapter 6.

If the communications difficulties persist, use the secure access sequence, also described in chapter 6, to delete all of your existing CDCNET connections. Then, create a new connection to the service and check to see if the job you exited is recoverable.

Passthrough Service Difficulties

The following notes and cautions apply to the use of interactive terminal passthrough services:

- If you are using a passthrough service and need to interrupt data flow to enter a network command and work on some other task, you can use the escape to $NET connection sequence, described in chapter 6. This sequence is effective at all times, even for nontransparent input.

- If you use remote echoing, (common with VAX[1] systems), you might notice longer echoing delays when using passthrough connections than you would if you were directly connected to the echoing systems.

- When you issue commands to remote systems to abort output, notice that output is not terminated until all of the data in the passthrough connection has been displayed at the terminal. The amount of data varies.

- In a non-passthrough configuration, commands such as TRMDEF (NOS) can be used to set parity, X-ON/X-OFF flow control, and other parameters for the communication line that connects the terminal to CDCNET. However, if such commands are used in a passthrough environment, they affect the parameters of the remote system port connected to the passthrough communication line, not the terminal's parameters. When troubleshooting passthrough communications, the possibility of such commands embedded in user startup procedures on the remote system should be investigated.

  In limited circumstances it may be appropriate to initiate remote system communication line attribute changes in a passthrough configuration. However, you must fully understand the impact of doing this.

1. VAX is a registered trademark of the Digital Equipment Corporation.

- Before creating a passthrough connection for XMODEM transfers from a microcomputer, use the following command to disable parity and character flow control:

  ```
  CHATA P=NONE CFC=NO
  ```

  If you need to transfer files at speeds less than 9600 bits per second after you have connected to passthrough, take the following steps to set the TRANSPARENT_MESSAGE_LENGTH connection attribute to 135:

  1. Press the BREAK key or attention character (do not press RETURN).
  2. Hold down the CTRL key and press X (do not press RETURN).
  3. Enter the following string, followed by RETURN:

     ```
     CHACA TML=135 IEM=TRANSPARENT
     ```

Service Failures

It sometimes takes CDCNET software several minutes to inform interactive terminal users about the loss of their connections to NOS or NOS/VE host applications. Similar delays can be expected when the network informs a user of its inability to create new connections to NOS or NOS/VE host applications, and the inability to execute terminal user procedures using the DO command.

If your service does fail, and your terminal is disconnected from it, the network connects you to $NET and issues the following message:

```
You may enter CDCNET commands.
```

Try to access the service later using a CREATE_CONNECTION command. Or, use the DISPLAY_SERVICE command to display the latest status of your network's services.

Cursor Positioning Delays

Cursor positioning delays implemented by the host suspending output transmission cannot be expected to reliably pass through a network. The more networks there are between source and destination, the more likely this problem is to occur.

This problem is likely to be seen on an asynchronous passthrough connection when the host tries to perform cursor positioning delays by suspending output. The only reliable delays in the network environment are those accomplished by sending NUL characters for the desired time-delay interval.

Glossary A

Glossary A

A

Alphabetic Character

One of the following letters:

A through Z
a through z

American Standard Code for Information Interchange (ASCII)

A standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

ASCII

Refer to American Standard Code for Information Interchange.

Asynchronous Protocol

The communications protocol enabling the transmission of data as characters individually synchronized by start/stop bits.

Attribute

Refer to Terminal Attribute.

Auto Recognition

Refer to Automatic Recognition.

Automatic Recognition

The process whereby the network identifies attributes (characteristics) of the terminal when the communication line, connected to the terminal, becomes active.

B

Batch Mode

A mode of execution where a job is submitted and processed as a unit without intervention from the user.

Bit

Binary digit. A bit has the value of either 0 or 1.

Bits Per Second

A data transmission rating that expresses the flow of the smallest units of information per unit of time.

C

Catenet

Refer to Concatenated Network.

CCP

Refer to Communications Control Program.

CDCNET

Refer to Control Data Distributed Communications Network.

Character

(ISO) A member of a set of elements upon which agreement has been reached, and that is used for the organization, control, or representation of information. Characters may be letters, digits, punctuation marks, or other symbols. A character can be a graphic character or a control character.

Client

In a passthrough connection, the passthrough user that creates a connection to another passthrough user (the Server). See also Server.

Coded Character Set

(ISO) A set of unambiguous rules that establish a character set and the one-to-one relationships between the characters of the set and their coded representations.

Comment

Any character or sequence of characters (except the quotation mark ["]) that is preceded by a quotation mark. It is terminated by another quotation mark or the end of the physical line. The network treats a comment the same as it does a space.

Communication Line

A terminal line that establishes a complete communication circuit between a terminal and the network.

Communications Control Program (CCP)

Software that provides terminal access and remote processing capability for NOS.

Complete Transmission

Logical unit of data as processed by a service.

Concatenated Network (Catenet)

A communications network composed of more than one type of communications medium (more than one network solution); often established when it is necessary to interconnect a local area network (LAN) with other resources (for example, another local area network, or geographically remote computer-related resources). Also called a catenet.

Connection

1. (ISO) An association established between functional units for conveying information.

2. A data path between a terminal and a service through CDCNET. The default connection, $NET, handles communications between the terminal and CDCNET.

Connection Attributes

Characteristics unique to a particular connection, such as the type of editing to be done on input data. A terminal connected to CDCNET at any given time has as many sets of connection attributes as it has connections.

Control Character

(ISO) A character that occurs in a particular context and initiates, modifies, or stops a control operation. A control character is not a graphic character, but may have graphic representation.

Control Data Distributed Communications'/ Network (CDCNET)

1. The collection of compatible hardware and software products offered by Control Data to interconnect computer resources into distributed communications networks.

2. A network that is interconnected by Control Data Network Architecture (CDNA)-compatible hardware and software products.

CRC

Refer to Cyclic Redundancy Check.

Cursor

1. (ISO) A movable, visible mark used to indicate the position on which the next operation takes place on a display surface.

2. The pointer used by a terminal to indicate where the user is positioned on the screen.

Cursor Positioning

Placing the cursor to indicate where data should be entered or displayed.

Cyclic Redundancy Check (CRC)

A technique designed for bit-oriented checking of errors on a transmission basis. It uses a unique mathematical polynomial that is known to both the sender and the receiver.

D

Data Editing

Process of preparing data for a later operation. Editing may include the rearrangement or the addition of data, the deletion of unwanted data, etc.

Dedicated Line

A communication line that permanently connects a terminal to a device interface. Contrast with Switched Line.

Default

The assumed value for a parameter when the parameter is not specified by the user.

Device Interface (DI)

The communications processor that Control Data offers as its CDCNET hardware product. Also called a CDCNET device interface.

DI

Refer to Device Interface.

Digit

One of the following characters:

0 1 2 3 4 5 6 7 8 9

E

EBCDIC

Refer to Extended Binary Coded Decimal Interchange Code.

EIA Flow Control

The procedure of controlling the rate of data transfer with an EIA hardware interface.

Ellipsis

Two (or more) consecutive periods at the end of a physical line to indicate the continuation of a command. The ellipsis can be optionally preceded or followed by a space.

Escape to $NET Connection Sequence

The steps involved in switching from a connection in transparent mode to $NET.

Extended Binary Coded Decimal Interchange Code (EBCDIC)

A set of 256 characters, each represented by eight bits, that is used with the 3270 Binary Synchronous Communications protocol.

F

Field Attribute Character

Character in the first position of each display field in a formatted display of a 3270 display station.

Format Effectors

Any character used to control the positioning of printed or displayed data.

FSE

Refer to Full Screen Editor.

Full Screen Editor (FSE)

A text editor on NOS (called FSE) or NOS/VE (called EDIT_FILE) that allows you to edit files in either line mode or screen mode.

G

Graphic Character

A character that can be printed or displayed.

H

HASP Protocol

A job control protocol for transmitting data processing files and jobs between certain models of computers. It is also called the Houston Automatic Spooling Program.

I

IAF

Refer to Interactive Facility.

Input

Data flowing up-line from terminal to the service.

Interactive Facility (IAF)

An application used on NOS to provide a terminal user with interactive processing capability. The interactive facility makes terminal input/output and file input/output appear the same to an executing program.

Interactive Mode

A mode of execution where a user enters commands at a terminal and each command elicits a response from the computing service.

Interface

A mechanism that enables the exchange of data between two dissimilar resources in a communications network.

L

LAN

Refer to Local Area Network.

LIM

Refer to Line Interface Module.

Line Folding

The continuation of an output line that exceeds the terminal's width on the following line of the screen or paper.

Line Interface Module (LIM)

A smaller logic board within a CDCNET device interface that enables the device interface to be attached to terminal, workstation, and unit record equipment lines.

Local Area Network (LAN)

A privately owned communications network that interconnects computer-related resources. Typically, the resources interconnected by this network are confined to a relatively concise geographic area, such as a single building.

M

M-Bit

Refer to More Bit.

Microcomputer

A computer system whose processing unit is a microprocessor. A basic microcomputer includes a microprocessor, storage, and input/output facility.

Microprocessor

An integrated circuit that accepts and executes coded instructions, and delivers signals describing its status. The instructions may be entered, integrated, or stored internally.

Mode 4

A data communications protocol, consisting of variants 4A, 4B, and 4C. The Mode 4 protocol supports two-way alternate communications (where messages may be sent in one direction or another, but not in both directions simultaneously) on switched or dedicated synchronous lines within a line speed range of 1200 to 19200 bits-per-second.

The CDCNET Mode 4 terminal interface program supports the 4A and 4C variants of the Mode 4 protocol.

More Bit (M-Bit)

A data communications protocol flag that indicates whether there are more data packets in the message currently being delivered. If the M-Bit is set to TRUE, the message is incomplete.

N

NAM

Refer to Network Access Method.

NAM/VE

Refer to Network Access Method/Virtual Environment.

$NET

The connection between a user's terminal and CDCNET. The network reserves this connection for CDCNET commands, network responses, and messages from the network operator.

Network

An interconnected set of host computers, terminals, workstations, and unit record equipment. Refer also to Local Area Network and Concatenated Network.

Network Access Method (NAM)

The access method that resides under NOS; allows host-based network applications programs to exchange information with communications networks.

Network Access Method/Virtual Environment (NAM/VE)

The access method that resides under NOS/VE; allows host-based network applications programs to exchange information with communications networks.

Network Operating System (NOS)

The operating system for CYBER 170 computer systems.

Network Operating System/Virtual Environment (NOS/VE)

The virtual operating system for CYBER 180 computer systems.

Network Operator

A person who monitors CDCNET activity, has the ability to control CDCNET hardware and software, makes occasional network configuration changes, and performs elementary troubleshooting.

NOS

Refer to Network Operating System.

NOS/VE

Refer to Network Operating System/Virtual Environment.

O

Output

Data flowing from the service to the terminal.

P

Packet Assembly/Disassembly (PAD)

Facility that allows nonpacket-mode asynchronous, start-stop mode terminals to communicate with hosts accessible through an X.25 network.

PAD

Refer to Packet Assembly/Disassembly.

Parameter Name

A name that uniquely identifies a parameter.

Partial Transmission

Incomplete logical unit of data as processed by a service.

Passthrough

Refer to Terminal Passthrough.

PDN

Refer to Public Data Network.

Personal Computer

Refer to Microcomputer.

Protocol

A set of conventions that must be followed to achieve complete communications between the computer-related resources in a network.

Public Data Network (PDN)

A publicly-owned communications network that interconnects computer-related resources. Typically, the resources interconnected by this network are not confined to a relatively concise geographic area (such as an X.25 public data network).

R

Reprieve Processing

The process a program performs after receiving an interrupt from the network.

S

SCL

Refer to System Command Language.

Secure Access Sequence

A sequence of keystrokes that can be entered from an interactive terminal to delete all existing CDCNET service connections for that terminal. This sequence can be used to ensure there is a bona fide $NET connection before the terminal user begins to enter service user validations or other guarded information.

Server

In a passthrough connection, the passthrough user that defines a passthrough title for other passthrough users (clients) to connect to. See also Client.

Service

An entity that is external to CDCNET but is registered within CDCNET as being capable of conducting input and output with a terminal or with another service.

Site Administrator

A person who configures a site's network, supervises hardware and software changes, and ensures that the network operates efficiently.

Slave Printer

Although terminals can have multiple input or output mechanisms on a connection at any time, CDCNET is aware of only one. Such mechanisms operate in series and are called slave devices.

SNA

Refer to Systems Network Architecture.

String

A value that represents a sequence of characters.

Switched Line

A communication line connected with one device interface, but able to be connected to any one of several terminals via a switching mechanism, such as a dialed telephone line. Contrast with Dedicated Line.

System Command Language (SCL)

User interface to the CDC Network Operating System/Virtual Environment (NOS/VE).

Systems Network Architecture (SNA)

IBM standard defining the layers and layer protocols to be used within an IBM network.

T

TDP

Refer to Terminal Definition Procedure.

Terminal

Any interactive device used to access a service through CDCNET. It may be a microcomputer, batch console, video-display terminal, hardcopy terminal, display station, or workstation that can be used to conduct a dialog with your service.

Terminal Attributes

Characteristics that apply to all of a terminal's connections. A terminal connected to CDCNET has only one set of terminal attributes.

Terminal Definition Procedure (TDP)

An optional configuration file that defines a terminal device or devices connected to a line whenever the line becomes active. A TDP can be used to define a terminal device that differs from the default terminal device type defined by the TIP that controls the line.

Terminal Interface Program (TIP)

A program that provides an interface for terminals connected to a device interface supporting terminals. TIPs provide default line and terminal configurations. Optional terminal definition procedures (TDPs) and terminal user procedures (TUP) can be used to define terminals having attributes that differ from those provided by the TIP's defaults.

Terminal Passthrough

A CDCNET feature that allows interactive asynchronous terminal traffic to pass through the network transparently. The hosts and terminals interface with each other as if they were directly connected. Terminal passthrough allows a CDCNET-connected terminal user to access non-CDCNET supported hosts, such as NOS/BE and VAX.

Terminal Redefinition Command

A command issued by a service or application that changes terminal or connection attributes.

Terminal User Procedure (TUP)

An optional configuration file that defines attributes of terminals and connections. A TUP can be used to define attributes for a particular terminal model or a group of terminals. A TUP for a terminal is executed when the communication line from the terminal to the supporting device interface becomes active.

TIP

Refer to Terminal Interface Program.

Transparent Mode

A way of handling data in which the network performs little or no input editing and does not format or translate output. The INPUT_EDITING_MODE terminal attribute designates input in this mode as transparent.

Transparent Mode Escape Sequence

The steps involved in switching from NORMAL input editing mode while engaged in a Transparent mode session.

TUP

Refer to Terminal User Procedure.

Type-Ahead

The ability of a terminal user to enter input even though the service has not finished processing the user's previous input(s).

V

Virtual Line Mode

A way of handling data in which the network edits input, and formats and translates output. The INPUT_EDITING_MODE terminal attribute designates input in this mode as normal.

W

Working Connection

The path a terminal uses to exchange information with a computing service or the network. It is one of the user's existing connections. If the user is communicating directly with the network, the connection is $NET.

X

X.PC

An asynchronous data communications protocol that improves the networking capabilities of personal computers. It also allows users to have multiple active virtual circuits.

X.25

The Consultative Committee of International Telephone and Telegraph (CCITT) standard for the interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) in an X.25 packet-switching network.

X.29 PAD

A CDCNET feature that allows asynchronous terminals to access CDCNET either by a Public Data Network (PDN) that supports the X.3 Packet Assembly/Dissassembly (PAD) facility or by the terminals operating in X.25 mode.

3

3270 Bisynchronous TIP

A terminal interface program that provides support for the IBM 3270 Information Display System. The 3270 Bisynchronous TIP allows 3271, 3274, 3275, and 3276 control units to connect directly to CDCNET in order to communicate with a CDCNET terminal device interface (TDI) over dedicated or dial-up lines using the centralized multipoint Binary Synchronous Communication protocol. The 3270 TIP Bisynchronous supports up to 32 multi-dropped clusters of up to 32 devices on each line.

3270 SNA Communications TIP

A terminal interface program that provides IBM 3270 Information Display System users access to CDCNET through an SNA network.

ASCII Coded Character Set B

ASCII Coded Character Set B

This appendix contains the complete ASCII coded character set, in table B-1. This appendix also describes how you can change an attribute setting by assigning one or more ASCII characters to the attribute.

Three groups of attributes can be set using the methods described in this appendix. Within this guide, each attribute in the first two groups is described in terms of the key you press to perform an action. Your terminal or communications package determines how the action is performed.

- The first group consists of terminal attributes that you can change using a single ASCII character. The following terminal attributes belong to this group.

  ATTENTION_CHARACTER (AC)
  BACKSPACE_CHARACTER (BC)
  BEGIN_LINE_CHARACTER (BLC)
  CANCEL_LINE_CHARACTER (CLC)
  END_LINE_CHARACTER (ELC)
  END_PARTIAL_CHARACTER (EPC)
  NETWORK_COMMAND_CHARACTER (NCC)

- The second group contains connection attributes that you can change using a list of ASCII characters. The following connection attributes belong to this group.

  TRANSPARENT_FORWARD_CHARACTER (TFC)
  TRANSPARENT_TERMINATE_CHARACTER (TTC)

- The third group consists of terminal attributes that you can change using a sequence of characters. The sequence defines the series of characters the network sends your terminal when the action occurs.

  CARRIAGE_RETURN_SEQUENCE (CRS)
  END_OUTPUT_SEQUENCE (EOS)
  FORM_FEED_SEQUENCE (FFS)
  LINE_FEED_SEQUENCE (LFS)

Changing an Attribute

If you need to change an attribute, the following procedure outlines the steps you take.

1. Decide which characters you can assign the attribute.

   If you want to change an attribute in the second group (TFC, TTC) or one in the third group (CRS, EOS, FFS, LFS), you can use any ASCII character in the ASCII coded character set (see table B-1).

   If you want to change an attribute in the first group (AC, BC, BLC, CLC, ELC, EPC, NCC), refer to the table specified in the following list.

Terminal Attribute	Available Characters
AC	See table B-2.
BC	See table B-3.
BLC	See table B-2.
CLC	See table B-4.
ELC	See table B-5.
EPC	See table B-4.
NCC	See table B-6.

2. Select an appropriate character (or characters).

3. Display your current terminal attributes by issuing this command:

   ```
   %DISTA
   ```

   If you are going to use TRANSPARENT mode, also display your connection attributes.

   ```
   %DISCA
   ```

4. Review the attributes listed by DISPLAY TERMINAL ATTRIBUTES (and DISPLAY_CONNECTION_ATTRIBUTES). By verifying that the new character you have selected is not already assigned to another character attribute, you can avoid producing unexpected results. Repeat steps 2, 3, and 4 as necessary.

5. Refer to the appropriate table to find out how you can represent this character (as a graphic, mnemonic, character string, control character, or code radix).

6. Enter a CHANGE_TERMINAL_ATTRIBUTE or CHANGE_CONNECTION_ATTRIBUTE command to change the attribute.

Using Examples

The following examples illustrate how you can enter ASCII characters, sequences, and lists.

Entering ASCII Characters

If you change CLC from the cancel mnemonic (CAN) to the exclamation point (!), you can enter the attribute in the following ways.

Entry	Example
Character string (enclosed in apostrophes)	`CLC='!'`
Decimal code	`CLC=33` or `CLC=33(10)`
Hexadecimal code	`CLC=21(16)`

If you want to change CLC from an exclamation point (!) back to CAN, you can enter it in the following ways.

Entry	Example
Mnemonic	`CLC=CAN`
Control character (entered by simultaneously pressing the CONTROL key and specified graphic key)	`CLC=^X`
Decimal code	`CLC=24` or `CLC=24(10)`
Hexadecimal code	`CLC=18(16)`

Using NUL or an empty string ('') deletes any character except the BACKSPACE_CHARACTER, END_LINE_CHARACTER, and NETWORK_COMMAND_CHARACTER. For instance, if you do not want a CANCEL_LINE_CHARACTER, set the attribute to NUL or an empty string.

```
CLC=NUL
CLC=''
```

When you enter a DISPLAY_TERMINAL_ATTRIBUTE command to verify the setting, the network displays NUL regardless of whether you entered NUL or an empty string. The network uses NUL for these characters because NUL does not have any meaning for most terminals.

```
Cancel_Line_Character      :NUL
```

Entering Sequences

You can reset the third group of attributes by entering sequences as shown in the following EOS examples. To separate a sequence of more than one character, use spaces or commas, and enclose the sequence in parentheses.

```
EOS=(BEL '/')
EOS=(BEL,'/')
```

In addition to mnemonics and strings, you can also use decimal codes, hexadecimal codes, and control characters in sequences.

```
EOS=(^G '/')
EOS=(7 47)
EOS=(7(16) 2F(16))
```

When you specify a sequence of characters for an attribute, *all* the characters in the sequence are used by the network to perform the action.

You can use NUL or an empty string ('') to delete a sequence of characters.

```
EOS=NUL
EOS=''
```

When you display the setting, the network does not show a value.

```
END_OUTPUT_SEQUENCE        :
```

However, if you enter a NUL character in a sequence of two or more characters, the network will accept it as a value.

Entering Lists

You enter lists for the second group of attributes in the same way you do sequences. As shown in the following examples, a list is also separated by spaces or commas and enclosed in parentheses.

```
TTC=(CR ETX EOT RS)
TTC=(CR,ETX,EOT,RS)
```

This TTC list contains four characters. Unlike characters in a sequence, when you specify a list of characters for an attribute, the network recognizes any *one* of the characters listed.

Instead of using NUL or an empty string (''), you nullify a list of characters by specifying NONE for TRANSPARENT_CHARACTER_MODE.

```
TCM=NONE
```

Table B-1. Complete ASCII Coded Character Set (Characters Available for CRS, EOS, FFS, LFS, TFC, and TTC)

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
000	00	NUL	Null	^@
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
017	11	DC1	Device control 1 (X-ON)	^Q or ^q
018	12	DC2	Device control 2	^R or ^r
019	13	DC3	Device control 3 (X-OFF)	^S or ^s
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_

(Continued)

Table B-1. Complete ASCII Coded Character Set (Characters Available for CRS, EOS, FFS, LFS, TFC, and TTC) *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
032	20	SP	Space	
033	21	!	Exclamation point	
034	22	"	Quotation marks	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	
039	27	'	Apostrophe	
040	28	(	Opening parenthesis	
041	29	)	Closing parenthesis	
042	2A	*	Asterisk	
043	2B	+	Plus	
044	2C	,	Comma	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
048	30	0	Zero	
049	31	1	One	
050	32	2	Two	
051	33	3	Three	
052	34	4	Four	
053	35	5	Five	
054	36	6	Six	
055	37	7	Seven	
056	38	8	Eight	
057	39	9	Nine	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
061	3D	=	Equals	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
065	41	A	Uppercase A	
066	42	B	Uppercase B	
067	43	C	Uppercase C	

(Continued)

Table B-1. Complete ASCII Coded Character Set (Characters Available for CRS, EOS, FFS, LFS, TFC, and TTC) *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
068	44	D	Uppercase D	
069	45	E	Uppercase E	
070	46	F	Uppercase F	
071	47	G	Uppercase G	
072	48	H	Uppercase H	
073	49	I	Uppercase I	
074	4A	J	Uppercase J	
075	4B	K	Uppercase K	
076	4C	L	Uppercase L	
077	4D	M	Uppercase M	
078	4E	N	Uppercase N	
079	4F	O	Uppercase O	
080	50	P	Uppercase P	
081	51	Q	Uppercase Q	
082	52	R	Uppercase R	
083	53	S	Uppercase S	
084	54	T	Uppercase T	
085	55	U	Uppercase U	
086	56	V	Uppercase V	
087	57	W	Uppercase W	
088	58	X	Uppercase X	
089	59	Y	Uppercase Y	
090	5A	Z	Uppercase Z	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
097	61	a	Lowercase a	
098	62	b	Lowercase b	
099	63	c	Lowercase c	
100	64	d	Lowercase d	
101	65	e	Lowercase e	
102	66	f	Lowercase f	
103	67	g	Lowercase g	

(Continued)

Table B-1. Complete ASCII Coded Character Set (Characters Available for CRS, EOS, FFS, LFS, TFC, and TTC) *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
104	68	h	Lowercase h	
105	69	i	Lowercase i	
106	6A	j	Lowercase j	
107	6B	k	Lowercase k	
108	6C	l	Lowercase l	
109	6D	m	Lowercase m	
110	6E	n	Lowercase n	
111	6F	o	Lowercase o	
112	70	p	Lowercase p	
113	71	q	Lowercase q	
114	72	r	Lowercase r	
115	73	s	Lowercase s	
116	74	t	Lowercase t	
117	75	u	Lowercase u	
118	76	v	Lowercase v	
119	77	w	Lowercase w	
120	78	x	Lowercase x	
121	79	y	Lowercase y	
122	7A	z	Lowercase z	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	
127	7F	DEL	Delete	

Table B-2. ASCII Characters Available for AC and BLC

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
000	00	NUL	Null	^@
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
018	12	DC2	Device control 2	^R or ^r
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_
033	21	!	Exclamation point	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	
039	27	'	Apostrophe	

(Continued)

Table B-2. ASCII Characters Available for AC and BLC *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
042	2A	*	Asterisk	
043	2B	+	Plus	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	

Table B-3. ASCII Characters Available for BC

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
018	12	DC2	Device control 2	^R or ^r
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_
033	21	!	Exclamation point	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	
039	27	'	Apostrophe	
042	2A	*	Asterisk	

(Continued)

Table B-3. ASCII Characters Available for BC *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
043	2B	+	Plus	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	
127	7F	DEL	Delete	

Table B-4. ASCII Characters Available for CLC and EPC

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
000	00	NUL	Null	^@
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
018	12	DC2	Device control 2	^R or ^r
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_
033	21	!	Exclamation point	
034	22	"	Quotation marks	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	

(Continued)

Table B-4. ASCII Characters Available for CLC and EPC *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
039	27	'	Apostrophe	
040	28	(	Opening parenthesis	
041	29	)	Closing parenthesis	
042	2A	*	Asterisk	
043	2B	+	Plus	
044	2C	,	Comma	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
061	3D	=	Equals	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	

Table B-5. ASCII Characters Available for ELC

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
018	12	DC2	Device control 2	^R or ^r
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_
033	21	!	Exclamation point	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	
039	27	'	Apostrophe	
042	2A	*	Asterisk	

(Continued)

Table B-5. ASCII Characters Available for ELC *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
043	2B	+	Plus	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	

Table B-6. ASCII Characters Available for NCC

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
001	01	SOH	Start of heading	^A or ^a
002	02	STX	Start of text	^B or ^b
003	03	ETX	End of text	^C or ^c
004	04	EOT	End of transmission	^D or ^d
005	05	ENQ	Enquiry	^E or ^e
006	06	ACK	Acknowledge	^F or ^f
007	07	BEL	Bell	^G or ^g
008	08	BS	Backspace	^H or ^h
009	09	HT	Horizontal tabulation	^I or ^i
010	0A	LF	Line feed	^J or ^j
011	0B	VT	Vertical tabulation	^K or ^k
012	0C	FF	Form feed	^L or ^l
013	0D	CR	Carriage return	^M or ^m
014	0E	SO	Shift out	^N or ^n
015	0F	SI	Shift in	^O or ^o
016	10	DLE	Data link escape	^P or ^p
018	12	DC2	Device control 2	^R or ^r
020	14	DC4	Device control 4	^T or ^t
021	15	NAK	Negative acknowledge	^U or ^u
022	16	SYN	Synchronous idle	^V or ^v
023	17	ETB	End of transmission block	^W or ^w
024	18	CAN	Cancel	^X or ^x
025	19	EM	End of medium	^Y or ^y
026	1A	SUB	Substitute	^Z or ^z
027	1B	ESC	Escape	^[
028	1C	FS	File separator	^\
029	1D	GS	Group separator	^]
030	1E	RS	Record separator	^^
031	1F	US	Unit separator	^_
033	21	!	Exclamation point	
034	22	"	Quotation marks	
035	23	#	Number sign	
036	24	$	Dollar sign	
037	25	%	Percent sign	
038	26	&	Ampersand	
039	27	'	Apostrophe	

(Continued)

Table B-6. ASCII Characters Available for NCC *(Continued)*

Decimal Code	Hex Code	Graphic/ Mnemonic	Name or Meaning	Control Character
040	28	(	Opening parenthesis	
041	29	)	Closing parenthesis	
042	2A	*	Asterisk	
043	2B	+	Plus	
044	2C	,	Comma	
045	2D	-	Hyphen	
046	2E	.	Period	
047	2F	/	Slant	
058	3A	:	Colon	
059	3B	;	Semicolon	
060	3C	<	Less than	
061	3D	=	Equals	
062	3E	>	Greater than	
063	3F	?	Question mark	
064	40	@	Commercial at	
091	5B	[	Opening bracket	
092	5C	\	Reverse slant	
093	5D	]	Closing bracket	
094	5E	^	Circumflex	
095	5F	_	Underline	
096	60	`	Grave accent	
123	7B	{	Opening brace	
124	7C	|	Vertical line	
125	7D	}	Closing brace	
126	7E	~	Tilde	

Messages C

Messages C

This appendix alphabetically lists informative and diagnostic messages that the network sends to your terminal.

Understanding Messages

Each message listing contains two parts:

Message — The message that the network sends is listed in boldface, and any variable within that message is shown in italics.

Description — A description follows each message. It explains the status or problem and identifies any variable in the message. In addition, it describes any action you should take.

If you receive a message that is not listed in this appendix, report it in one of the following ways:

- Ask your site administrator to write a Programming System Report (PSR).
- Note the message on the comment sheet at the back of this manual.

List of Messages

This alphabetic list contains the messages you receive at your terminal. Messages beginning with symbols (such as dashes) and with variables appear at the end of the list.

ACTXPC command not allowed.

The TIP defined for this terminal does not support the X.PC protocol.

Alphabetic character in number: *integer.*

You entered the value shown as an integer in a network command. Integers used as values in network commands must conform to the syntax described in chapter 2.

If you are entering a radix, use digits within parentheses immediately after the integer. If your use of radix is correct, ensure that the integer does not contain a letter out-of-range for its radix.

Attribute name *attribute* **is invalid.**

The network did not recognize the attribute you specified on the DISPLAY_TERMINAL_ATTRIBUTE or DISPLAY_CONNECTION_ATTRIBUTE command.

Look for a typographical error in the attribute name or abbreviation. Check the command format, and reenter the command.

Attributes changed.

The network successfully executed the CHANGE_TERMINAL_ATTRIBUTE or CHANGE_CONNECTION_ATTRIBUTE command that you just entered.

Auto recognition not defined for this line.

You entered the ACTIVATE_AUTO_RECOGNITION command. However, the network cannot perform automatic recognition for your terminal because the site administrator has not configured your communication line for it.

If you need to change PARITY or CODE_SET, use a CHANGE_TERMINAL_ATTRIBUTE command. (Chapter 4 describes the attributes). If you need to change line speed, contact your site administrator.

Cannot create connection.

The network cannot find the service name you requested. Try again later.

Cannot create name starting with $.

The connection name in the CREATE_CONNECTION command just entered began with a dollar sign ($). Connection names beginning with a dollar sign are reserved for internal CDCNET use. Reenter the command with a unique connection name that does not begin with a dollar sign.

Cannot delete connection $NET.

Only ASYNCTIP or X25_ASYNCTIP users can delete the $NET connection with the DELC $NET command. If you are not using either of these two TIPs, or you did not specify a CONNECTION_NAME parameter value of $NET and your working connection is $NET, the network returns this response.

Cannot locate service ***service name.***

The network cannot connect you to the service you requested on the CREATE_CONNECTION command due to one of the following:

- The network does not recognize the service name because it contains a typographical error.
- The service is not currently connected through CDCNET.

If the name is spelled correctly, either reenter the command later or select another comparable service. If the name is misspelled, reenter the command using the correct spelling.

Character code integer is out of range.

An integer value for a character is out of range. The network accepts integers within the range of 1 .. FF(16). (See appendix B for information on entering decimal and hexadecimal codes.) Reenter the command with an appropriate integer value.

Character code name ***value*** **is invalid.**

The control code mnemonic specified in your command is not valid. Table B-1 lists the valid mnemonics for control characters NUL through US. Reenter the command using a valid mnemonic.

Character code name *value* is too long.

The specified portion of your last command contains more characters than are allowed. The command contains a typographical or syntax error. If you attempted to enter a hexadecimal value, ensure that the first digit is 0 through 7. Reenter the command using the correct format.

Character code string is too long.

The string that you entered as a character value was greater than one character long. (See appendix B for information on using a string as a character value.)

Client connection complete system=*system_name* line=*line_name*.

A client, on the named system with the named line, has paired with your passthrough server connection.

Command only valid for X.25 terminals.

A SET_PAD_MESSAGE (SETPM) command was entered from a non-X.25 terminal.

Command *entry* not allowed from $NET.

Since you are connected to $NET, you cannot enter a DISPLAY_CONNECTION_ATTRIBUTE or CHANGE_CONNECTION_ATTRIBUTE command. Enter a CHANGE_WORKING_CONNECTION or CREATE_CONNECTION command to access a service connection and reenter your original command.

Command *entry* too long.

The command cited is longer than 256 characters. Work with your site administrator to shorten the command before executing the procedure again.

Conflict in *attribute* and *attribute:*

You attempted to change an attribute listed in the message. The attempt failed due to one of the following:

- The attributes cannot have the same values.
- You assigned an incorrect value to one of the attributes.

Reenter the CHANGE_TERMINAL_ATTRIBUTE or CHANGE_CONNECTION_ATTRIBUTE command with a different value for the attribute you want changed. (See appendix B for information on entering character values.)

Conflict in TFC and AC.

The network sends this message, preceded by the message 'No attributes changed.' in response to a CHACA command. The value just declared for the TRANSPARENT_FORWARD_CHARACTER is already in use as the ATTENTION_CHARACTER. When CDCNET receives that value, it processes the value as an ATTENTION_CHARACTER only. If that usage is not acceptable, you should change one of the two attributes.

Conflict in TFC and CFC.

The network sends this message, preceded by the message 'No attributes changed.' in response to a CHACA command. The TRANSPARENT_FORWARD_CHARACTER cannot be DC1 or DC3 when CHARACTER_FLOW_CONTROL is set to ON. With this setting, CDCNET processes only DC1 or DC3 as a flow control character. If that usage is not acceptable, you should change the TRANSPARENT_FORWARD_CHARACTER.

Conflict in TTC and AC.

The network sends this message, preceded by the message 'No attributes changed.' in response to a CHACA command. The value just declared for the TRANSPARENT_TERMINATE_CHARACTER is already in use as the ATTENTION_CHARACTER. When CDCNET receives that value, it processes it as an ATTENTION_CHARACTER only. If that usage is not acceptable, you should change one of the two attributes.

Conflict in TTC and CFC.

The network sends this message, preceded by the message 'No attributes changed.' in response to a CHACA command. The value just declared for the TRANSPARENT_TERMINATE_CHARACTER cannot be DC1 or DC3 when CHARACTER_FLOW_CONTROL is set to ON. With this setting, CDCNET processes only DC1 or DC3 as a flow control character. If that usage is not acceptable, you should change the TRANSPARENT_TERMINATE_CHARACTER.

Conflict in TTC and TFC.

The network sends this message, preceded by the message 'No attributes changed.' in response to a CHACA command. The value just declared for the TRANSPARENT_TERMINATE_CHARACTER is already in use as the TRANSPARENT_FORWARD_CHARACTER. When CDCNET receives this value, it processes it as the TRANSPARENT_FORWARD_CHARACTER only. If that usage is not acceptable, you should change one of the two attributes.

Connection to Passthrough Service complete.

You created a connection to the passthrough service. You are now able to enter a DEFINE_PASSTHROUGH_TITLE command.

Connection to Site Passthrough service ***service name*** **complete.**

You entered a CREATE_CONNECTION command and successfully created a connection to a passthrough service that was defined by your site administrator. The second line of this message identifies the DI system and line that you are using.

System=*service name*, **Line=***line name*.

Connection to User Passthrough service ***service name*** **complete.**

You entered a CREATE_CONNECTION command and successfully created a connection to a passthrough service that was defined by another user. The second line of this message identifies the DI system and line that you are using.

System=*service name*, **Line=***line name*.

Connection ***connection name*** **created.**

The network successfully executed the CREATE_CONNECTION command you just entered or your connection that was waiting has completed.

Connection *connection name* **deleted.**

The network successfully executed the DELETE_CONNECTION command you just entered.

Connection *connection name* **pending.**

The connection name you specified in the CHANGE_WORKING_CONNECTION has not been changed because it is pending. This message is issued when you use CREATE_CONNECTION command with WAIT=YES.

Connection *connection name* **is unknown.**

The connection name you specified in the CHANGE_WORKING_CONNECTION or DELETE_CONNECTION command contained a typographical error or did not exist. As a result, CDCNET cannot identify the correct connection.

Reenter the command if the original problem was a typographical error. If not, enter a DISPLAY_CONNECTIONS command to see if the connection still exists.

Connection *connection name* to *service_name* **complete.**

The connection that was pending has successfully been created.

Copyright Control Data Corporation, 1985, 1986, 1987, 1988.

The network sends a four-line message when you access CDCNET. In addition to the copyright information, the network sends the following three lines to your terminal. The variable parts of the message are indicated in the following example by the letter x.

```
DI System Name is 080025xxxxxx, xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Terminal Name is xxxxxx, xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
You may enter CDCNET commands.
```

The second line gives the DI system name, the third identifies your terminal name, and the fourth prompts you to enter network commands.

If you need assistance with your connection, knowing the DI system name and terminal name can help the site administrator or network operator respond effectively. (See the DISPLAY_CONNECTIONS command in chapter 2 for more information on these names.)

Digit too large for radix of integer: *integer*

A digit within the integer is greater than or equal to the radix value. (See chapter 2 for information on using radixes.)

Duplicate connection name.

The CREATE_CONNECTION command that you just entered specified a name of an existing connection.

Use the DISPLAY_CONNECTIONS command to check the names of your existing connections. Reenter the CREATE_CONNECTION command with a unique connection name.

ENGAGED

The network issues this message after receiving an X.28 command requesting the status of the X.25 connection. The message indicates an X.25 connection exists with a remote DTE.

Expecting command, found *entry.*

The network expected a command instead of what it found. Your entry did not adhere to network syntax; a delimiter is probably missing. (See chapter 2 for information on command syntax.)

Expecting digit in number.

The network found an alphabetic or special character instead of an integer value.

Expecting end of default specification for parameter *entry* **found** *value.*

The specified text occurred immediately following the default specification and is not meaningful there. A delimiter was probably omitted. Ask your site administrator to write a PSR.

Expecting end of expression, found *'entry'.*

The entry was found following an expression that was apparently meant to be part of that expression, but could not be interpreted as such. You probably omitted a delimiter or operator. Correct the expression.

Expecting end of parameter *entry* found *value*.

The specified entry occurred immediately following the default specification and is not meaningful there. You probably omitted a delimiter. Check the parameter format, and reenter it.

Expecting end of value for parameter *entry* found *value*.

The specified entry occurred immediately following a value for the parameter and is not meaningful there. You probably omitted a delimiter. Check parameter format, and reenter it.

Expecting end of value list for parameter *entry*.

The command you just entered contained an incorrect parameter. Check command format, and reenter the command.

Expecting end of value set for parameter *entry*.

The specified entry occurred immediately following a value set for the parameter and is not meaningful there. You probably omitted a delimiter. Check parameter format, and reenter it.

Expecting keyword, found *entry*.

The network expected a keyword value, but instead found the entry cited (or empty). The names that many of the terminal attributes and connection attributes accept as values are also called keyword values. For example, the names INPUT_EDITING_MODE uses (NORMAL and TRANSPARENT) are keyword values.

Expecting *type-1*, **found** *type-2*.

The command you entered contains the wrong type of parameter value. Type-1 indicates which of the following values the network expected.

- Value
- Boolean
- Name
- Integer
- Character

Instead of finding that type of value, the network encountered one of the following (type-2).

- End of expression
- End of line

This type-2 option means you may have entered an incomplete command or not pressed the RETURN key (not sent an END_LINE_CHARACTER). Reenter the command using the correct format.

Expecting value, found @.

The procedure name parameter on the DO command begins with the @ character. These names are valid only for TUPs that are executed as a result of a network definition command (such as DEFINE_LINE).

FREE

The network issues this message after receiving an X.28 STAT command, requesting the status of the X.25 connection. Message indicates no X.25 connection with a remote DTE currently exists.

Improper parameter description table: NAMES is NIL but PARAMETERS is not.

The network detected an error in the command's parameter description table: no parameter names are specified but parameter values are defined. Ask your site administrator to write a PSR.

Improper parameter description table: ^NAMES [i].NUMBER > UPPERBOUND(PARAMETERS^).

The network detected an error in the command's parameter description table: the number of parameter names is greater than the number of parameter definition items. Ask your site administrator to write a PSR.

Improper parameter description table: Not all PARAMETERS have a NAME.

The network detected an error in the command's parameter description table: a parameter definition item exists with no name. Ask your site administrator to write a PSR.

Improper parameter description table: PARAMETERS is NIL but NAMES is not.

The network detected an error in the command's parameter description table: no parameter value definitions exist but parameter names are defined. Ask your site administrator to write a PSR.

Improper parameter description table: Too many names.

The network detected an error in the command's parameter description table: the number of possible parameter values defined in the command's parameter description table exceeds the maximum allowed number of parameter values. Ask your site administrator to write a PSR.

Improper radix specifier: ***entry*****.**

The network found a bad radix specifier in the command string. (See chapter 2 for information on using radixes.)

Improper radix value: ***entry*****.**

The network found an incorrect value in the radix; it must be 8, 10, or 16. (See chapter 2 for information on using radixes.)

Input cancelled.

You performed an action that caused CDCNET to discard your input. Using a CANCEL_LINE_CHARACTER or possibly pressing an ATTENTION_CHARACTER key causes such a condition. Reenter your input.

Input discarded.

The network discarded your last command since you previously entered too many CDCNET commands that the network has not yet processed.

Insufficient resources to load X.PC TIP.

No RAM is available on the Communications Interface Module (CIM) to load the X.PC TIP CIM software. Add another CIM for the X.PC device or remove an unrequired TIP from the CIM associated with this device.

Invalid user interrupt.

The network returns this response when you enter a Network Command character followed by a character that is not alphabetic or numeric, but a reserved symbol. Only alphabetic or numeric characters may follow a Network Command character.

Integer value *value* is out of range.

The network detected an integer value outside the expected range (as defined in the command's parameter description table).

Invalid value specified for parameter *-entry-*.

The attribute that you specified on the CHANGE_TERMINAL_ATTRIBUTE or CHANGE_CONNECTION_ATTRIBUTE command does not apply to the protocol (such as Asynchronous or HASP) you are using. Therefore, you cannot change it.

Missing radix specifier for integer: ***integer*.**

The integer cannot contain letters unless it is a hexadecimal value (with A, B, C, D, E, or F); it must begin with a digit and end with the radix specification (16). Reenter the command and enclose the radix in parentheses.

Missing string delimiter: ***'entry*.**

An apostrophe, which delimits strings, was missing from the string cited. Reenter the command enclosing the string in apostrophes.

Name value *value* too long.

The name you specified on the command was longer than the allowed 31 characters. For example, you may have entered a connection name on the CREATE_CONNECTION command that was too long. Reenter the command using the correct format.

Name value *value* too short.

The network detected a name value shorter than expected (as defined in the command's parameter description table).

Network is currently congested.

The network cannot process the CREATE_CONNECTION command you just entered because it lacks the resources now to support another connection. Try again later. Deleting unneeded connections may help relieve the congestion.

Network operator may have disabled requests.

If your command was REQUEST_NETWORK_OPERATOR, the operator is not receiving messages from terminal users. Contact your site administrator to enable alarm message 168 if you need this capability.

No attributes changed.

The message following this one explains why your attributes remain unchanged. Correct the problem and try again. The message also precedes a second message identifying an attribute conflict.

No attributes changed. Conflict in use of control code *control code*.

The control or replacement code used is not permitted or out of range.

No connection present -- timeout.

You did not create a connection within the allowed time and have been disconnected from the network. Reaccess CDCNET and create a connection when the network prompts you to enter CDCNET commands. (See Understanding Configuration Options in chapter 1 for information on timeouts or contact your site administrator to find out the length of your site's disconnect timeout.)

No displayable services in list.

When you enter a DISPLAY SERVICE command, the network sends this message if your site administrator has not specified any service names to be displayed.

No services are displayable.

When you enter a DISPLAY_SERVICE command, the network sends this message if your network administrator has not permitted the display of any services at your terminal.

Not a required operator device of an I/O station.

The CREATE_CONNECTION (CREC) command was rejected because the requested service is a C170 Batch Gateway title and the console from which the CREC command was entered is *not* the required operator device of a batch I/O station. A console can be defined as the required operator device of a batch I/O station using the DEFINE_I_O_STATION (DEFIOS) configuration command.

Output discarded.

The network discarded output sent to your terminal on $NET since the connection's queue is already full.

Parameter CONNECTION_NAME is required when DELC is entered from the $NET connection.

The network returns this response when you enter the DELC command from the $NET connection without specifying a CONNECTION_NAME parameter value.

Parameter *entry* given more than once.

The command that caused this message contained duplicate parameters. The network ignored the command and stopped executing the procedure.

Look for a typographical error in a parameter name or abbreviation. Check the command format, and change the command. Rerun the procedure.

Parameter *entry* is required but was omitted.

The command that caused this message does not contain the required parameter cited. The network ignored the command and stopped executing the procedure. Look for a missing blank or comma. Check the command format, and change the command by adding the missing parameter. Rerun the procedure.

PAR INV:*n*

The network issues this message after receiving an X.25 Terminal Gateway command (SET) specifying an invalid X.3 reference number.

PAR*n*:INV

The network issues this message after receiving an X.25 Terminal Gateway command (SET) specifying an invalid value for an X.3 reference number.

Passthrough client connection disconnected.

The passthrough service is notifying you, the passthrough server, that the paired client has disconnected.

Passthrough connection attributes reset.

By entering the $NET-switch sequence (BREAK key or attention character, CONTROL key, and C), you escaped to $NET from your working connection (a passthrough service). Subsequently, you entered a CHANGE_WORKING_CONNECTION command and returned to the passthrough-service connection. When you pressed the RETURN key (entered an END_LINE_CHARACTER) after returning, the network reset your connection attributes with passthrough connection attribute values.

Passthrough connection *connection name* timeout.

Since you did not enter any data during the last 30 seconds, the passthrough service has disconnected your passthrough connection (with the connection name identified in the message) from the passthrough server you were using.

Passthrough connection ***connection name*** **timeout in 30 seconds.**

After issuing this message, the network sends an ASCII BEL character. To avoid being disconnected, enter data within 30 seconds.

Passthrough input discarded. Please define titles.

If you enter input other than a DEFINE_PASSTHROUGH_TITLE command after a CREATE_CONNECTION command to your site's passthrough service, the network discards the input and prompts you to enter the command.

Passthrough input discarded. Passthrough connection attributes reset.

By entering the escape to $NET connection sequence, you escaped to $NET from your working connection (a passthrough service). Subsequently, you entered a CHANGE_WORKING_CONNECTION command and returned to the passthrough-service connection. The network reset your connection attributes and discarded the data you entered before pressing the RETURN key.

Passthrough input discarded. Waiting for client.

After you are configured as a passthrough server, all your input is discarded until a client connection is paired with your server connection.

Passthrough server connection disconnected.

The passthrough service is notifying you, the passthrough client, that the paired server connection has disconnected and, as a result, the network will now disconnect your connection.

Passthrough service stopped.

Both the server and the client are notified with this message when a network operator stops the passthrough service.

Passthrough service titles defined.

In response to your DEFINE_PASSTHROUGH_TITLE command, the network has registered the names you specified.

Passthrough service titles not registered.

The network sends this message if it cannot register, in the CDCNET directory, the titles specified by your DEFINE_PASSTHROUGH_TITLE command. Reenter the DEFPT command.

Passthrough service titles registered.

The network sends this message after registering the titles specified by your DEFINE_PASSTHROUGH_TITLE command.

Passthrough titles not defined.

You must enter a DEFINE_PASSTHROUGH_TITLE command again in order to configure yourself as a passthrough server.

PROC statement missing or not first.

Either the PROC statement is missing, or it is not the first statement in the procedure. Ask your site administrator to change the first statement to an appropriate PROC statement. When the problem has been corrected, execute the procedure again.

Procedure name mismatch. DO *procedure name* **and PROC** *procedure name.*

In the procedure you just tried to execute, the procedure name in the PROC statement does not match the procedure name you specified on the DO command. Correct the problem by doing one of the following:

- Check for typographical errors and reenter the DO command with the correct procedure name.
- Work with your site administrator to make the procedure names specified on the PROC statement and the DO command match.

Procedure nesting limit reached.

A terminal user procedure can only contain one DO command for another terminal user procedure. Work with your site administrator to reduce the number of nested procedures to one.

Procedure service is unavailable.

The CDCNET software that processes requests to execute terminal user procedures is temporarily unavailable. If you received this message in response to a DO command, try reentering the command later. If it appeared when you initially accessed the service, your automatically executed terminal user procedure did not execute. In either case, you are currently using the default terminal attributes and connection attributes.

Procedure *procedure name* aborted.

The network did not successfully execute the cited terminal user procedure. It ignored some or all of the commands. Additional messages should identify the problem that needs fixing.

Procedure *procedure name* access error.

The network cannot read the procedure you just tried to execute. The problem may be a non-ASCII88 formatted procedure file.

Procedure *procedure name* completed.

The network executed the cited procedure successfully. This procedure was called either automatically when you accessed your service or by a DO command that you just entered.

Procedure *procedure name* not found.

Execution of the named procedure was requested, either automatically at initial connection to CDCNET, or by using a DO command. Since this procedure is not available to CDCNET, it did not execute the procedure.

Work with your site administrator to solve this problem. CDCNET may not be able to find the procedure because the site administrator has not yet stored it, or because the procedure name is misspelled.

Request forwarded to network operator.

The network successfully executed the REQUEST_NETWORK_OPERATOR command that you just entered.

Searching for service *service name.* **Connection** *connection_name* **pending.**

The service you asked to be connected to is unavailable, and the connection is pending. This message is issued when you use CREATE_CONNECTION command with WAIT=YES.

Service *service name* **busy.**

The service cannot accept another connection at this time. Either reenter the command at a later time or select another service. This message is received when you use CREATE_CONNECTION with WAIT=NO.

Service *service name* **unavailable.**

You cannot use the service specified for one of the following reasons:

- The service cannot accept another connection at the moment.
- You are not allowed access to that service.

Either reenter the command at a later time or select another comparable service. If the name is misspelled, reenter the command using the correct spelling.

SET ERR

The SET command contains a syntax error. Correct the syntax and reenter the command.

String value too long.

The network detected a string value longer than expected (as defined in the command's parameter description table). For example, the string in a PUT_STRING command contained more than 80 characters. Shorten the string and reenter the command.

String value too short.

The network detected a string value shorter than expected (as defined in the command's parameter description table).

Table overflow: pvt names: insufficient memory to parse command.

An error occurred when the network tried to allocate global memory space for the parameter names. Reenter the command later, or contact your site administrator.

TIP rejects connection.

The TIP that provides the interface to this terminal cannot accept this connection. For example, the X.PC TIP supports multiple logical channels over a single communications link and the TIP controls the total number of user connections on the link.

Table overflow: pvt parameters: insufficient memory to parse command.

An error occurred when the network tried to allocate global memory space for the parameters. Reenter the command later, or contact your site administrator.

Table overflow: pvt values: insufficient memory to parse command.

An error occurred when the network tried to allocate global memory space for the parameter values. Reenter the command later, or contact your site administrator.

Table overflow: string values: insufficient memory to parse command.

Memory to hold a string value was requested from the network but not satisfied. Reenter the command later when there is less congestion, or contact your site administrator.

Too few values given for parameter *entry*.

The network found too few values in the parameter. Reenter the command using the correct format.

Too few values in value set given for parameter *entry*.

The network found too few values in the value set in the parameter. Reenter the command using the correct format.

Too few values sets given for parameter *entry*.

The network found too few value sets in the parameter. Reenter the command using the correct format.

Too many parameters given.

More parameters were entered with the command than are defined for the command. This usually occurs when parameters are given positionally or when a command has no parameters. Reenter the command using the correct format.

Too many values given for parameter *entry*.

The network found too many values in the parameter. Reenter the command using the correct format.

Too many values in value set given for parameter *entry*.

The network found too many values in the value set in the parameter. Reenter the command using the correct format.

Too many value sets given for parameter *entry*.

The network found too many value sets in the parameter. Reenter the command using the correct format.

Unable to process ACTXPC commands (no resources).

The network currently lacks the resources to process the ACTXPC command. Try again later.

Unable to process command *entry*.

The network lacks the resources to process that command at this time. Try again later.

Unexpected call to *entry*.

The command processor processing a command made an unexpected call to one of the parser routines. Ask your site administrator to write a PSR.

Unexpected *entry* after command.

The specified entry immediately followed a command name. You probably omitted a delimiter. Reenter the command.

Unknown command *entry*.

The indicated command does not exist. Check the spelling of the command name or abbreviation, and reenter the command.

Unknown destination for put string.

The PUT_STRING command attempted to send data to a service when no connection to it existed. Establish a connection with the CREATE_CONNECTION command before executing the procedure again.

User connection limit exceeded.

You attempted to create a new connection with a valid CREATE_CONNECTION command, but you have already reached your connection limit. (See Understanding Configuration Options in chapter 1 for more information on this limit.)

You can display your existing connections and connection limit with the DISPLAY_CONNECTIONS command. Delete a connection with the DELETE_CONNECTION command, and then reenter the CREATE_CONNECTION command.

User interrupt ignored.

You have already either pressed the BREAK key or the ATTENTION_CHARACTER key or have entered a %1 or %2. The network can process only one of these at a time.

Value range not allowed for parameter *entry*.

The network does not recognize value ranges (such as 1..6) in the parameter cited. Reenter the command using the correct format.

Working connection changed to *connection name*, **service name** *service name*.

The network successfully executed the CHANGE_WORKING_CONNECTION command that you just entered. You are communicating on the named working connection. The service on that connection is available for you to use.

You may enter CDCNET commands.

You are connected to $NET, the connection that the network uses to communicate with your terminal. You can enter CDCNET commands without prefixing them with the NETWORK_COMMAND_CHARACTER. Either disconnect from CDCNET or enter a command (such as CREATE_CONNECTION, CHANGE_WORKING_CONNECTION, or another CDCNET command).

--ERROR-- An X25 connection currently exists.

The network issues this response after receiving a CREATE_X25_CONNECTION command from a terminal that already has an active X25 connection.

--ERROR-- A remote_dte_address may include only digits 0 through 9.

The network issues this message after receiving a CREXC command with an invalid REMOTE_DTE_ADDRESS parameter value. The REMOTE_DTE_ADDRESS parameter of the CREATE_X25_CONNECTION command must contain only digits 0 through 9. Reenter the command with a valid REMOTE_DTE_ADDRESS.

--ERROR-- Connection broken. The local DTE has cleared the connection. Clearing cause code=XX and diagnostic code=YY.

The network issues this message after the local DTE has broken the X.25 connection. Compare the hexadecimal clearing cause code and the diagnostic code in the message with table values in the CCITT X.25 specifications to determine why the local DTE cleared the connection.

--ERROR-- Connection broken. The remote DTE has cleared the connection. Clearing cause code=XX and diagnostic code=YY.

The network issues this message after the remote DTE has broken the X25 connection. Compare the hexadecimal clearing cause code and the diagnostic code in the message with table values in the CCITT X.25 specifications to determine why the remote DTE cleared the connection.

--ERROR-- Connection broken. The X.25 interface has been stopped by a network operator.

The network issues this message when the operator breaks the X.25 physical connection.

--ERROR-- Connection broken. The X.25 link has temporarily gone down.

The network issues this message when the X.25 PDN goes down.

--ERROR-- Connection broken. The X.25 link is currently experiencing problems.

The network issues this message when it detects an error in the X.25 PDN, indicated by an X.25 restart packet. The problem may be a congested network.

--ERROR-- Connection broken. The X.25 link is now inoperative.

The network issues this message when an X.25 trunk goes down during an interactive connection.

--ERROR-- Connection rejected. All X.25 logical channels are currently busy.

The network issues this message if it cannot establish an X.25 connection because all X.25 logical channels are currently busy.

--ERROR-- Connection rejected. The local dte address is currently inoperative.

The network issues this message when it cannot establish an X.25 connection because the X.25 trunks are down.

--ERROR-- Connection rejected. The network is currently congested.

The network issues this message when it cannot create a connection because the NDI memory is congested.

--ERROR-- Connection rejected. No memory is available.

The network issues this message when an X.25 connection cannot be established because the device interface does not have enough memory to create the control blocks needed for the connection.

--ERROR-- Connection to X.25 Terminal Gateway broken. The network operator has stopped the X.25 Terminal Gateway.

The network issues this message after the network operator has stopped activity through an active X.25 Terminal Gateway. The network disconnects you from the X.25 Terminal Gateway and breaks the connection with the remote DTE with which you were communicating.

--ERROR-- Parameter COMMAND is required but was omitted.

The network issues this message after receiving a command with no COMMAND parameter.

--ERROR-- Parameter SERVICE_NAME is required but was omitted.

The CREXC command must include a SERVICE_NAME parameter value matching a site-defined title. Reenter the command.

--ERROR-- Too many parameters given.

The network issues this message when you enter a command with too many parameters. Reenter the command with the correct parameters.

--ERROR-- Unable to locate service ***service_name*****.**

The network issues this message when it cannot find the service specified by the CREXC command.

--ERROR-- Unauthorized remote DTE address specified.

The CREATE_X25_CONNECTION command specified a remote DTE address for which you are not authorized access.

--ERROR--*xxxx* is not a command.

The command name you specified is not a valid command.

speed **bps,** *code set,* **parity:** *parity.*

As a result of the automatic recognition process just completed, CDCNET is indicating the values used for line speed, CODE_SET, and PARITY. The terminal line speed is given as bits per second. CODE_SET, and PARITY are identified by attribute values (see chapter 3 for a description of the CODE_SET and PARITY terminal attributes).

If these values are acceptable, no action is needed. For more information on automatic recognition, see chapter 8 and the description of the ACTIVATE_AUTO_RECOGNITION command in chapter 2.

entry **is not a parameter name.**

The indicated portion of the last command you entered was interpreted as a parameter name. No such parameter exists for that command. Check for a typographical error or a misplaced space. Reenter the command using the correct format.

entry **is not an allowed keyword value.**

The network expected a keyword value, but instead found the entry cited (or empty). The names that many of the terminal attributes and connection attributes accept as values are also called keyword values. For example, the names INPUT_EDITING_MODE uses (NORMAL and TRANSPARENT) are keyword values.

remote DTE banner and login prompt

The network sends the banner and login prompt of the remote DTE after establishing a valid X.25 connection.

X.PC already activated.

The network returns this message after receiving an ACTXPC command, although the XPC TIP is currently active.

X.PC not defined for this line.

The X.PC TIP has not been defined in the DI connected to this device.

X.PC protocol being activated.

The network returns this message after successfully executing the ACTXPC command.

Migrating from CCP D

Migrating from CCP D

This appendix provides the following service-specific information for NOS and NOS/VE users:

- Briefly describes NOS/VE and NOS commands that manipulate CDCNET terminal attributes and connection attributes.
- Discusses migrating to CDCNET from CCP, the communication control program that resides in a 255X communication device.

This service-specific information is based on the assumption you are accessing a NOS/VE or NOS service through CDCNET. That is, you are processing on one of the following:

- NOS/VE on a CYBER 180 mainframe. (You are using NAM/VE and CDCNET.)
- NOS on a CYBER 170 mainframe or a CYBER 180 mainframe that runs NOS-only or NOS and NOS/VE in dual state. (You are using NAM and CDCNET.)

Using NOS/VE and NOS Commands

Services can provide commands that enable you to change, display, and use terminal attributes and connection attributes.

NOS/VE supports the following:

CYBIL procedure calls — On NOS/VE, you can use CYBIL procedure calls to manipulate CDCNET attributes. These include the following:

IFP$CHANGE_TERM_CONN_ATTRIBUTES
IFP$CHANGE_TERM_CONN_DEFAULTS
IFP$CHANGE_TERMINAL_ATTRIBUTES
IFP$FETCH_TERM_CONN_ATTRIBUTES
IFP$GET_TERM_CONN_ATTRIBUTES
IFP$GET_TERM_CONN_DEFAULTS
IFP$GET_TERMINAL_ATTRIBUTES
RMP$REQUEST_TERMINAL
IFP$STORE_TERM_CONN_ATTRIBUTES

See CYBIL File Management Usage for descriptions of these calls.

SCL commands — On NOS/VE, you can use SCL commands to manipulate CDCNET attributes. These include the following:

CHANGE_CONNECTION_ATTRIBUTES
CHANGE_TERM_CONN_DEFAULTS
CHANGE_TERMINAL_ATTRIBUTES
DISPLAY_CONNECTION_ATTRIBUTES
DISPLAY_TERM_CONN_DEFAULTS
DISPLAY_TERMINAL_ATTRIBUTES
DISPLAY_TERMINAL_MODEL
SET_TERMINAL_ATTRIBUTES

See NOS/VE System Usage for descriptions of SCL commands that change and retrieve CDCNET attributes.

NOS supports the following:

TRMDEF command	You may use either CDCNET attributes or CCP mnemonics with the TRMDEF command. However, you cannot use both in the same command. Mnemonics are also used with the CCP terminal definition commands. In this manual, they are called terminal definition mnemonics. For example, CI is used for carriage-return idle count and CN for cancel character. (See NOS Version 2 Reference Set, Volume 3, System Commands for descriptions of this command and the mnemonics.)
Control bytes	In a program, you can use field number/field value pairs (FN/FV) with the 0010 (terminal redefinition for NAM/CDCNET) and 0016 (terminal redefinition) control byte to alter the characteristics of a terminal connected to the Interactive Facility (IAF) on NOS. (See NOS Version 2 Reference Set, Volume 4, Program Interface for more information.)
CTRL/CHAR/R supervisory message	In an application (such as the Interactive Facility [IAF] and Printer Support Utility [PSU]), you can use field number/field value pairs (FN/FV) with the CTRL/CHAR/R supervisory message to redefine the characteristics of a terminal. The application sends this message to the Network Access Method (NAM). (See Network Products, Network Access Method [NAM] Version 1, Host Application Programming Reference Manual for more information.)

Migrating from CCP to CDCNET

If you have been using CCP and are now communicating with your service through CDCNET, the change may affect your use of terminal characteristics in the following ways.

- Using the CDCNET default Network Command Character
- Using CDCNET attributes with the TRMDEF command
- Using terminal definition mnemonics with the TRMDEF command
- Effects of changing NOS terminal classes
- CDCNET mapping of field number/field value (FN/FV) pairs

Using the CDCNET Default Network Command Character

When accessing NOS through CDCNET, you must be aware that the default Network Command Character (NCC) is a percent sign (%) and not the ESCAPE key (ESC). You can, if you prefer, change the NCC by using the CDCNET command, CHANGE_TERMINAL_ATTRIBUTE (CHATA). See chapters 2 and 3 for more information on CHATA and NCC respectively.

Using CDCNET Attributes with the TRMDEF Command

When accessing NOS through CDCNET, you can specify CDCNET terminal attributes instead of terminal definition mnemonics. To do this, enter terminal attributes in the $tc_i=v_i$ parameter of the TRMDEF command.

Using attributes enables you to manipulate a larger set of terminal characteristics than you can with terminal definition mnemonics. However, you must adhere to NOS conventions and TRMDEF requirements when you specify attributes. The following examples illustrate the differences between entering terminal attributes on the CHANGE_TERMINAL_ATTRIBUTE (CDCNET) and TRMDEF (NOS) commands.

Difference	Examples

Integer codes	CDCNET accepts decimal and hexadecimal codes as values for integers. Hexadecimal codes must include a trailing radix (16).
	`%CHATA CLC=21(16)`
	The TRMDEF command requires an X before a hexadecimal value.
	`TRMDEF,CLC=X21`

Empty or null	CDCNET and the TRMDEF command represent empty strings differently.
	CDCNET uses two apostrophes.
	`%CHATA EOS=''`
	The TRMDEF command accepts only empty strings for the following terminal attributes:
	CARRIAGE_RETURN_SEQUENCE
	END_OUTPUT_SEQUENCE
	FORM_FEED_SEQUENCE
	LINE_FEED_SEQUENCE
	The command recognizes the following format as an empty string.
	`TRMDEF,EOS=.`

Character delimiters	CDCNET uses apostrophes to delimit characters.
	`%CHATA CLC='!'`
	The TRMDEF command uses dollar signs.
	`TRMDEF,CLC=$!$`

Difference	**Examples**

Sequences	CDCNET uses sequences for the following terminal attributes:
	CARRIAGE_RETURN_SEQUENCE
	END_OUTPUT_SEQUENCE
	FORM_FEED_SEQUENCE
	LINE_FEED_SEQUENCE
	The TRMDEF command allows you to enter hexadecimal codes for these sequences and TERMINAL_MODEL.
	`TRMDEF,EOS=X072F`
	CDCNET accepts hexadecimal codes as values for sequences.
	`%CHATA EOS=(7(16) 2F(16))`
	In addition, CDCNET recognizes decimal codes, strings, mnemonics, and control characters as values.
	`%CHATA EOS=(7 47)`
	`%CHATA EOS=(BEL '/')`
	`%CHATA EOS=(^G '/')`

Difference	Examples

Lists	CDCNET uses lists for the following connection attributes:
	
	TRANSPARENT_FORWARD_ CHARACTER
	TRANSPARENT_TERMINATE_ CHARACTER
	
	CDCNET recognizes strings, decimal codes, hexadecimal codes, mnemonics, and control characters as values in lists.
	
	`%CHACA TTC=('a' 'b' ^G)`
	`%CHACA TTC=(97 98 BEL)`
	`%CHACA TTC=(61(16) 62(16) 7(16))`
	
	When you verify the entries by entering a DISPLAY_TERMINAL_ATTRIBUTE command, the values are displayed in lowercase only if they are entered in lowercase. (Likewise, values entered in uppercase are displayed in uppercase.) The displays always show the real value for an attribute.
	
	With the TRMDEF command, you can enter a list of characters as strings with slashes (/) alone or surrounded by dollar signs ($).
	
	`TRMDEF,TTC=A/B/C`
	`TRMDEF,TTC=$A$/$B$/$C$`
	
	When you enter uppercase characters in these TRMDEF formats, the network displays the values in uppercase. If you try to enter lowercase characters, NOS changes the characters to uppercase before sending them to CDCNET.
	
	To use lowercase characters, enter hexadecimal codes.
	
	`TRMDEF,TTC=X61/X62/X63`

Difference	Examples

Use of ASCII characters	CDCNET allows you to enter all character values as a decimal code, hexadecimal code, character string, mnemonic, or control character. (See appendix B.)
	
	The TRMDEF command requires you to enter the ASCII codes represented by decimal values 1 through 31 as octal or hexadecimal values.
	
	`TRMDEF,CLC=30B`
	`TRMDEF,CLC=X18`
	
	Unlike CDCNET, the TRMDEF command does not accept a character represented as a mnemonic or control character. For example, CDCNET accepts the following entries, while the TRMDEF command does not.
	
	`%CHATA CLC=CAN`
	`%CHATA CLC=^X`

Using Terminal Definition Mnemonics with the TRMDEF Command

If you continue to use terminal definition mnemonics with the TRMDEF command, the network converts them into terminal attributes and connection attributes. However, they are not one-to-one correspondences.

Corresponding Terminal Attributes and Connection Attributes

Table D-1 shows which attributes CDCNET sets when you enter terminal definition mnemonics as TRMDEF parameters. Due to space considerations, the attributes listed are abbreviated.

TRANSPARENT Input

TRMDEF commands may contain one or more values for a DL or XL parameter. Table D-1 shows the change(s) CDCNET makes when you enter a TRMDEF command with a specified DL or XL value. If CDCNET does not establish a value for TRANSPARENT_CHARACTER_MODE, TRANSPARENT_LENGTH_MODE, or TRANSPARENT_TIMEOUT_MODE in this conversion process, the network automatically sets that connection attribute to NONE.

For example, suppose you enter the following TRMDEF command with XL set to C1 to specify the transmission length.

```
TRMDEF,XL=C1
```

In response, CDCNET sets two connection attributes for XL=C1.

TRANSPARENT_LENGTH_MODE=FORWARD
TRANSPARENT_MESSAGE_LENGTH=1

The network also sets the following connection attributes because the TRMDEF command did not specify XL=Xxx and XL=TO.

TRANSPARENT_CHARACTER_MODE=NONE
TRANSPARENT_TIMEOUT_MODE=NONE

Table D-1. Changing CDCNET Attributes with a TRMDEF Command

Terminal Definition Mnemonics	Corresponding Terminal Attributes and Connection Attributes
AB	Ignored.
BF=0	PCF=ON
BF=1	PCF=OFF, IBS=100
BF=2	PCF=OFF, IBS=200
BS=character	BC=character
B1	Ignored.
B2	Ignored.
CI=integer	CRD=integer*linespeed factor
CN=character	CLC=character
CP=Y	ELP=LFS, EPP=CRS
CP=N	ELP=NONE, EPP=NONE
CT=character	NCC=character
DL=Xxx	TCM=T, TFC=character, TTC=character, TLM=N, TTM=N
DL=Ccount	TLM=T, TML=integer, TCM=N, TTM=N
DL=TO	TTM=T, TCM=N, TLM=N
EB	Ignored.
EL=character	ELC=character
EL=EB	Ignored.
EL=EL	Ignored.
EL=CR	ELP=CRS
EL=LF	ELP=LFS
EL=CL	ELP=CRSLFS
EL=NO	ELP=NONE
EP=Y	E=ON
EP=N	E=OFF

(Continued)

Table D-1. Changing CDCNET Attributes with a TRMDEF Command *(Continued)*

Terminal Definition Mnemonics	Corresponding Terminal Attributes and Connection Attributes
FA=Y	EPC=NUL, SBC=ON, SND=ON
FA=N	EPC=LF, SBC=OFF, SND=OFF
IC=Y[1]	CFC=ON
IC=N[1]	CFC=OFF
IN=BK	Ignored.
IN=KB	IEM=N
IN=PT	IEM=N
IN=X	IEM=T
IN=XK	IEM=T
IN=XP	IEM=T
LI=integer	LFD=integer*linespeed factor
LK=Y	SA=D
LK=N	SA=S
OC=Y	CFC=ON
OC=N	CFC=OFF
OP=DI	FL=OFF
OP=PR	FL=ON
OP=PT	Ignored.
PA=E	P=EVEN
PA=N	P=NONE
PA=O	P=ODD
PA=Z	P=ZERO
PA=I	P=NONE
PG=Y	HP=ON
PG=N	HP=OFF
PL=integer	PL=integer
PW=integer	PW=integer

1. If both the IC and OC terminal definition mnemonics are specified, the last setting the network encounters takes precedence.

(Continued)

Table D-1. Changing CDCNET Attributes with a TRMDEF Command *(Continued)*

Terminal Definition Mnemonics	Corresponding Terminal Attributes and Connection Attributes
SE=Y	CLC=NUL, EPC=NUL, SBC=ON
SE=N	CLC=CAN, EPC=LF, SBC=OFF
XL=Xẋx	TCM=FT, TFC=character, TTC=character, TLM=N, TTM=N
XL=C1	TLM=F, TML=1, TTM=N, TCM=N
XL=Ccount	TLM=FE, TML=integer, TTM=N, TCM=N
XL=TO	TTM=T, TLM=N, TCM=N

Effects of Changing NOS Terminal Classes

On NOS, there are three ways to change terminal class:

- TRMDEF command with the TC terminal definition command
- 0016 control byte via a program
- CTRL/CHAR/R supervisory message to NAM via an application (such as IAF or PSU)

Whenever your NOS terminal class is changed by any of these methods, CDCNET automatically changes certain attributes. (CDCNET does not support auto input, the marking of output so that it is returned to the service with input.)

CDCNET sets the following terminal attributes and connection attributes for *all* terminal class changes.

Type of Attribute	Attribute Settings
Terminal	CANCEL_LINE_CHARACTER=CAN CHARACTER_FLOW_CONTROL=ON END_LINE_CHARACTER=CR END_LINE_POSITIONING=LFS END_PARTIAL_CHARACTER=LF END_PARTIAL_POSITIONING=CRS HOLD_PAGE=OFF NETWORK_COMMAND_CHARACTER=% PARITY=EVEN STATUS_ACTION=SEND
Connection	BREAK_KEY_ACTION=0 PARTIAL_CHARACTER_FORWARDING=OFF STORE_BACKSPACE_CHARACTER=OFF STORE_NULS_DELS=OFF TRANSPARENT_CHARACTER_ MODE=TERMINATE TRANSPARENT_FORWARD_CHARACTER=CR TRANSPARENT_LENGTH_ MODE=TERMINATE TRANSPARENT_MESSAGE_LENGTH=2043 TRANSPARENT_TIMEOUT_MODE=NONE

In addition to these global effects, CDCNET changes some terminal-attributes settings based on the specified terminal class. Table D-2 shows these settings for the classes that the network supports. (The network does not support terminal class 4.)

Table D-2. Selected Terminal Attribute Settings for NOS Terminal Classes 1, 2, 3, 5, 6, 7, and 8

	1	2	3	5	6	7	8
BC	BS	BS	BS	none	BS	BS	BS
CRD[1]	2	0	0	1	0	0	0
CRS	CR	CR	CR	ESC G	CR	CR	CR
E	OFF	OFF	OFF	OFF	OFF	ON	OFF
FFD	2	2	2	2	2	2	999ms
FFS	6 LFs	EM/ CAN	FF	ESC R	FS	ESC[H ESC[J	ESC FF
FL	ON	OFF	OFF	OFF	OFF	ON	OFF
LFD[1]	1	0	0	3	3	0	0
LFS	LF	LF	LF	ESC B	0	LF	LF
PL	0	24	30	24	27	24	35
PW	72	80	80	80	74	80	74

1. Millisecond (ms) value is dependent on line speed.

The remaining terminal attributes and connection attributes are *not* affected by terminal class changes.

Type of Attribute	Attributes
Terminal	ATTENTION_CHARACTER BEGIN_LINE_CHARACTER CODE_SET END_OUTPUT_SEQUENCE END_PAGE_ACTION HOLD_PAGE_OVER TERMINAL_MODEL
Connection	ATTENTION_CHARACTER_ACTION INPUT_BLOCK_SIZE INPUT_EDITING_MODE INPUT_OUTPUT_MODE TRANSPARENT_TERMINATE_CHARACTER

CDCNET Mapping of Field Number/Field Value Pairs

When you use one of the following, CDCNET converts the terminal characteristics that you specify as field number/field value pairs to attribute settings.

- IAF 0016 control byte in a NOS program
- CTRL/CHAR/R supervisory message that an application sends to NAM

Corresponding Terminal Attributes and Connection Attributes

Table D-3 shows the terminal attributes and connection attributes that the network uses when you specify field number/field value pairs in the following.

- 0016 control byte
- CTRL/CHAR/R supervisory message

Because of space considerations, the attribute settings and field names are abbreviated.

TRANSPARENT Input

Converting field number/field value pairs to CDCNET attributes for TRANSPARENT input is quite complex. You should send the following field number/field value pairs (decimal) together to ensure proper results.

56, 57, 58, 59, 60, 69, 70, and 146

CDCNET converts these field number/field value pairs into the TRANSPARENT_CHARACTER_MODE, TRANSPARENT_LENGTH_MODE, and TRANSPARENT_TIMEOUT_MODE settings, as shown in figures D-1 through D-5.

Table D-3. Mapping of FN/FV Pairs to CDCNET Attributes

Decimal Field Number	Octal Field Number	Field Name	Field Value	Corresponding Terminal Attributes and Connection Attributes
25	31	BF	0 1 2	PCF=ON PCF=OFF, IBS=100 PCF=OFF, IBS=200
30	36	XBZ1	all	Ignored.
31	37	XBZ2	all	Ignored.
32	40	LK	0 1	SA=S SA=D
33	41	HD	all	Ignored.
34	42	TC	all	See the preceding section, Effects of Changing NOS Terminal Classes.
35	43	PW	integer	PW=integer
36	44	PL	integer	PL=integer
37	45	PG	0 1	HP=OFF HP=ON

(Continued)

Table D-3. Mapping of FN/FV Pairs to CDCNET Attributes *(Continued)*

Decimal Field Number	Octal Field Number	Field Name	Field Value	Corresponding Terminal Attributes and Connection Attributes
38	46	CN	character	CLC=character
39	47	BS	character	BC=character
40	50	CT	character	NCC=character
41	51	AB	all	Ignored.
42	52	B1	all	Ignored.
43	53	B2	all	Ignored.
44	54	CI	integer	CRD=integer* linespeed factor
45	55	LI	integer	LFD=integer* linespeed factor
46	56	CA	1	Ignored.
47	57	LA	1	Ignored.
49	61	EP	0	E=OFF
			1	E=ON
50	62	PA	0	P=ZERO
			1	P=ODD
			2	P=EVEN
			3	P=NONE
			4	P=NONE
51	63	BR	0	BKA=0
			1	BKA=1
52	64	XPT	0	IEM=N
			1	IEM=T
53	65	IN	all	Ignored.

(Continued)

Table D-3. Mapping of FN/FV Pairs to CDCNET Attributes *(Continued)*

Decimal Field Number	Octal Field Number	Field Name	Field Value	Corresponding Terminal Attributes and Connection Attributes
54	66	OP	0 1 2	FL=ON FL=OFF Ignored.
55	67	FA	0 1	EPC=LF, SBC=OFF, ND=OFF EPC=NUL, SBC=ON, SND=ON
56	70	DL/XL	0 1	See figure D-1.
57	71	UBTCC	all	See figure D-4.
58	72	LBTCC	all	See figure D-4.
59	73	TDC	all	See figure D-2.
60	74	TTM	0 1	See figure D-5.
61	75	EL	character	ELC=character
62	76	ELO	all	Ignored.
63	77	CPEL	0 1 2 3	ELP=NONE ELP=CRS ELP=LFS ELP=CRSLFS
64	100	EB	all	Ignored.
65	101	EBO	all	Ignored.
66	102	CPEB	all	Ignored.

(Continued)

Table D-3. Mapping of FN/FV Pairs to CDCNET Attributes *(Continued)*

Decimal Field Number	Octal Field Number	Field Name	Field Value	Corresponding Terminal Attributes and Connection Attributes
67	103	IC[1]	0 1	CFC=OFF CFC=ON
68	104	OC[1]	0 1	CFC=OFF CFC=ON
69	105	MTIC	all	See figure D-3.
70	106	TMM	1	See figures D-1, D-4, and D-5.
71	107	CP	0 1	ELP=NONE, EPP=NONE ELP=LFS, EPP=CRS
87	127	FDLX	0 1	IOM=S[2] IOM=F
102	146	PP	character	EOS=character
112	160	NTA	0 1	IOM=U IOM=S[2]
146	222	STTO	0 1	See figure D-5.
147	223	CRI	integer	CRD=integer*4
148	224	LFI	integer	LFD=integer*4

1. If both the IC and OC field names are specified, the last field value the network encounters takes precedence.

2. Solicited INPUT_OUTPUT_MODE differs between CDCNET and CCP. CCP rejects input from the user if the service is not ready to accept input. CDCNET accepts input, but does not edit it until the service requests input.

```
If Transparent Input Character (56) is specified (1)
    Then if Transparent Message Mode (70) is specified as Multimessage (1)
        Then if Multimessage Transparent Input Character (69) is specified
            Then set TRANSPARENT_CHARACTER_MODE=FORWARD
        If not, set TRANSPARENT_CHARACTER_MODE=
        FORWARD-TERMINATE
    If not, set TRANSPARENT_CHARACTER_MODE=TERMINATE
If not, set TRANSPARENT_CHARACTER_MODE=NONE
```

Figure D-1. Corresponding TRANSPARENT_CHARACTER_MODE Settings

```
If Transparent Delimiter Character (59) is specified
    Then set TRANSPARENT_FORWARD_CHARACTER=(value of 59)
        If TRANSPARENT_FORWARD_CHARACTER=CR
            If the terminal's parity is IGNORE
                Then set TRANSPARENT_FORWARD_CHARACTER=
                (0D(16),8D(16))
            If not, leave TRANSPARENT_FORWARD_CHARACTER=CR
        If not, leave TRANSPARENT_FORWARD_CHARACTER=
        (value of 59)
If not, leave TRANSPARENT_FORWARD_CHARACTER unchanged
```

Figure D-2. Corresponding TRANSPARENT_FORWARD_CHARACTER Settings

```
If Multimessage Transparent Input Character (69) is specified
    Then set TRANSPARENT_TERMINATE_CHARACTER=(value of 69)
        If TRANSPARENT_TERMINATE_CHARACTER=CR
            If the terminal's parity is IGNORE
                Then set TRANSPARENT_FORWARD_CHARACTER=
                (0D(16),8D(16))
            If not, leave TRANSPARENT_TERMINATE_CHARACTER=CR
        If not, leave TRANSPARENT_TERMINATE_CHARACTER=
        (value of 69)
If not, leave TRANSPARENT_TERMINATE_CHARACTER unchanged
```

Figure D-3. Corresponding TRANSPARENT_TERMINATE_CHARACTER Settings

```
If Upper Byte Transparent Character Count (57) is specified and not 0
or Lower Byte Transparent Character Count (58) is specified and not 0
     Then set TRANSPARENT_MESSAGE_LENGTH=(value of 57 * 256) +
     (value of 58)
          If Transparent Message Mode (70) is specified as Multimessage (1)
               If TRANSPARENT_MESSAGE_LENGTH=1
                    Then set TRANSPARENT_LENGTH_MODE=FORWARD*
               If not, set TRANSPARENT_LENGTH_MODE=FORWARD_EXACT
          If not, set TRANSPARENT_LENGTH_MODE=TERMINATE
If not, set TRANSPARENT_LENGTH_MODE=NONE
* CCP forwards the exact length specified for all lengths including 1.
```

Figure D-4. Corresponding TRANSPARENT_LENGTH_MODE Settings

```
If Transparent Timeout (60) is specified (1)
     Then set TRANSPARENT_TIMEOUT_MODE=TERMINATE
          If Transparent Message Mode (70) is specified as Multimessage (1)
          and "Sticky" Transparent Timeout (146) is specified (1)
               Then set TRANSPARENT_TIMEOUT_MODE=FORWARD
          If not, stop
If not, set TRANSPARENT_TIMEOUT_MODE=NONE
```

Figure D-5. Corresponding TRANSPARENT_TIMEOUT_MODE Settings

Terminal Attribute Settings via TERMINAL_MODEL — E

Terminal Attribute Settings via TERMINAL_MODEL — E

If you are a NOS/VE or NOS user, there are several ways to change your terminal attribute settings. This appendix discusses how using each of the following commands affects these settings.

- DO command (CDCNET procedure call)
- CHANGE_TERMINAL_ATTRIBUTE (CDCNET) command
- NOS/VE commands
- NOS commands

The appendix also lists terminal attribute settings for selected terminals.

This information is based on the assumption that you are accessing a NOS/VE or NOS service through CDCNET. That is, you are processing on one of the following:

- NOS/VE on a CYBER 180 mainframe. (You are using NAM/VE and CDCNET.)
- NOS on a CYBER 170 mainframe or a CYBER 180 mainframe that runs NOS-only or NOS and NOS/VE in dual state. (You are using NAM and CDCNET.)

DO Command (CDCNET Procedure Call)

Both NOS/VE and NOS users can change their attributes with a CDCNET command. If you know your TERMINAL_MODEL, you can enter a DO command to change your terminal attributes. This command calls a terminal user procedure that contains the attribute settings for the specified terminal.

First, select the name from table E-1 that best suits your terminal. Next, issue a DO command using this name as the procedure name.

Table E-1. Procedure Names

Name	Class[1]	Your Terminal
CDC_721	3	CDC 721
CDC_722	2	CDC 722
CDC_722_30	2	CDC 722-30 running in ANSI mode
MAC_CONNECT_10	7	Macintosh computer using Control Data CONNECT version 1.0
MAC_CONNECT_11	7	Macintosh computer using Control Data CONNECT version 1.1
PC_CONNECT_10	7	IBM Personal Computer (or supported compatible) using Control Data CONNECT version 1.0
PC_CONNECT_11	7	IBM Personal Computer (or supported compatible) using Control Data CONNECT version 1.1
PC_CONNECT_12	7	IBM Personal Computer (or supported compatible) using Control Data CONNECT version 1.2

1. This column provides the NOS terminal class associated with each terminal.

(Continued)

Table E-1. Procedure Names *(Continued)*

Name	Class[1]	Your Terminal
PC_CONNECT_13	7	IBM Personal Computer (or supported compatible) using Control Data CONNECT version 1.3
PC_CONNECT_14	7	IBM personal Computer (or supported compatible) using Control Data CONNECT version 1.4
IBM_3270	-	IBM 3270
TV_955	-	Televideo 955
DEC_VT100	7	DEC VT100
DEC_VT220	7	DEC VT220
ZEN_Z19	(None)	Zenith Z19
ZEN_Z29	(None)	Zenith Z29

1. This column provides the NOS terminal class associated with each terminal.

When the network executes a DO command, it performs three actions:

- Changes your terminal attributes to the appropriate ones for the terminal you have selected. (See tables E-3 through E-7.)
- Changes the value of TERMINAL_MODEL and the other attributes in the DISPLAY_TERMINAL_ATTRIBUTES display.
- Sends appropriate information to the services on your connections. For a NOS/VE service, the TERMINAL_MODEL setting is sent. For a NOS service, the TERMINAL_MODEL setting is mapped into a terminal class, and the class is sent. What a service does with the information is service-dependent, but it should be able to handle this unsolicited message.

  NOS/VE uses this information to specify the correct terminal model for its EDIT_FILE utility. NOS does not use the information, but requires you to enter a SCREEN (NOS) command in addition to specifying a terminal model for its Full Screen Editor (FSE).

For example, the following DO command executes the DEC_VT100 procedure, changes your terminal attributes to the DEC_VT100 settings, and sends appropriate information to the services on your connections:

```
%DO DEC_VT100
```

Subsequently, when you issue a DISPLAY_TERMINAL_ATTRIBUTES command, it displays the same settings as shown in table E-6 for DEC_VT100.

CHANGE_TERMINAL_ATTRIBUTE (CDCNET) Command

The CHANGE_TERMINAL_ATTRIBUTE (CDCNET) command also enables you to change TERMINAL_MODEL regardless of whether you are working on NOS/VE or NOS.

If you are using the Asynchronous protocol, select a suitable name from table E-1 as the value of TERMINAL_MODEL. When you use a CHANGE_TERMINAL_ATTRIBUTE command to change TERMINAL_MODEL, the network performs the following actions.

- Changes the value of TERMINAL_MODEL in the DISPLAY_TERMINAL_ATTRIBUTES display.
- Does *not* change your terminal attributes (except for TERMINAL_MODEL) to those listed in tables E-3 through E-7.
- Sends appropriate information to the services on your connections. For a NOS/VE service, the TERMINAL_MODEL setting is sent. For a NOS service, the TERMINAL_MODEL setting is mapped into a terminal class, and the class is sent. What a service does with the information is service-dependent, but it should be able to handle this unsolicited message.

  NOS/VE uses this information to specify the correct terminal model for its EDIT_FILE utility. NOS does not use the information, but requires you to enter a SCREEN command to specify a terminal model for its Full Screen Editor (FSE).

For example, the following CHANGE_TERMINAL_ATTRIBUTE command changes the value of TERMINAL_NAME in the DISPLAY_TERMINAL_ATTRIBUTES display.

```
%CHATA TM=DEC_VT100
```

As a NOS user, this change does not affect other terminal attribute settings. As a NOS/VE user, it allows the EDIT_FILE utility to operate correctly with your terminal in screen mode, but it does not affect the terminal characteristics CDCNET uses.

If you are accessing a NOS service and using the HASP, Mode 4, or Binary Synchronous Communications protocol, you can change the TERMINAL_MODEL display with the CHANGE_TERMINAL_ATTRIBUTE command. Select an appropriate name for the TERMINAL_MODEL name from table E-2 and enter the command.

Table E-2. NOS TERMINAL_MODEL Names

Name	Class[1]	Your Terminal
IBM_HASP_POST	9	Terminal using the HASP protocol with printer devices supporting only postprint carriage control.
IBM_HASP_PRE	14	Terminal using the HASP protocol with printer devices supporting both postprint and preprint carriage control.
MODE4	15	Terminal using Mode 4A or 4C protocol.
IBM_3270	18	Terminal using the 3270 Binary Synchronous Communications protocol.

1. This column provides the NOS terminal class associated with each terminal.

NOS/VE Commands

If you are a NOS/VE user, other commands are available for changing TERMINAL_MODEL. In addition to the two CDCNET commands, DO and CHANGE_TERMINAL_ATTRIBUTE, you can use the following:

CHANGE_TERMINAL_ATTRIBUTE SET_TERMINAL_ATTRIBUTE	SCL commands for NOS/VE users (See NOS/VE System Usage for descriptions of SCL commands that change and retrieve CDCNET attributes.)
IFP$CHANGE_TERMINAL_ ATTRIBUTE	CYBIL procedure call (See CYBIL File Management Usage for descriptions of procedure calls.)

If you set TERMINAL_MODEL with either SCL command, NOS/VE uses the name for its EDIT_FILE utility, and sends the new TERMINAL_MODEL setting to CDCNET. Hence, the display produced by the DISPLAY_TERMINAL_ATTRIBUTES command reflects the change.

For example, the following SCL commands change the name used by the EDIT_FILE utility and the TERMINAL_MODEL setting displayed by CDCNET.

```
CHATA TM=DEC_VT100
SETTA TM=DEC_VT100
```

NOS Commands

On NOS, you can change TERMINAL_MODEL two additional ways. (See appendix D for more details.)

- Enter a TRMDEF command. Specify an appropriate name from table E-1 as the value of TERMINAL_MODEL. (See NOS Version 2 Reference Set, Volume 3, System Commands, for descriptions of this command and the mnemonics.)

```
TRMDEF,TM=DEC_VT100
```

- Use a 0016 control byte at the program level. (See NOS Version 2 Reference Set, Volume 4, Program Interface, for more information.)

Terminal Attribute Settings for Selected Terminals

Tables E-3 through E-7 list terminal attribute settings for the terminals named in table E-1. Use the CDCNET DO command, described in chapter 2, to set your terminal attributes to the values listed in these tables.

Table E-3. Selected Terminal Attribute Settings: CDC Terminals

Terminal Attribute	CDC 721	CDC 722	CDC 722_30
Attention_Character	NUL	NUL	NUL
Backspace_Character	BS	BS	BS
Begin_Line_Character	NUL	NUL	NUL
Cancel_Line_Character	CAN	CAN	CAN
Carriage_Return_Delay	0	0	0
Carriage_Return_Sequence	CR	CR	CR
Character_Flow_Control	ON	ON	ON
Code_Set	ASCII	ASCII	ASCII
Echoplex	OFF	OFF	OFF
End_Line_Character	CR	CR	CR
End_Line_Positioning	LFS	LFS	LFS
End_Output_Sequence			
End_Page_Action	NONE	NONE	NONE
End_Partial_Character	LF	LF	LF
End_Partial_Positioning	CRS	CRS	CRS
Fold_Line	OFF	OFF	OFF
Form_Feed_Delay	0	0	0
Form_Feed_Sequence	FF	EM CAN	ESC [H ESC [J
Hold_Page	OFF	OFF	OFF
Hold_Page_Over	ON	OFF	ON
Line_Feed_Delay	0	0	0
Line_Feed_Sequence	LF	LF	LF
Network_Command_Character	%	%	%
Page_Length	30	24	24
Page_Width	80	80	80
Parity	EVEN	EVEN	EVEN
Response_Action	SEND	SEND	SEND
Status_Action	SEND	SEND	SEND
Terminal_Model	CDC_721	CDC_722	CDC_722_30

Table E-4. Selected Terminal Attribute Settings: IBM Computers

Terminal Attribute	IBM Personal Computer
Attention_Character	NUL
Backspace_Character	BS
Begin_Line_Character	NUL
Cancel_Line_Character	CAN
Carriage_Return_Delay	0
Carriage_Return_Sequence	CR
Character_Flow_Control	ON
Code_Set	ASCII
Echoplex	OFF
End_Line_Character	CR
End_Line_Positioning	LFS
End_Output_Sequence	
End_Page_Action	NONE
End_Partial_Character	LF
End_Partial_Positioning	CRS
Fold_Line	ON
Form_Feed_Delay	0
Form_Feed_Sequence	ESC [H ESC [J
Hold_Page	OFF
Hold_Page_Over	ON
Line_Feed_Delay	0
Line_Feed_Sequence	LF
Network_Command_Character	%
Page_Length	24
Page_Width	80
Parity	EVEN
Response_Action	SEND
Status_Action	SEND
Terminal_Model	PC_CONNECT_xx[1]

1. With Control Data CONNECT version 1.0, set to 10; version 1.1, set to 11; version 1.2, set to 12; version 1.3, set to 13; version 1.4, set to 14.

Table E-5. Selected Terminal Attribute Settings: Macintosh

Terminal Attribute	Macintosh Computer
Attention_Character	NUL
Backspace_Character	BS
Begin_Line_Character	NUL
Cancel_Line_Character	CAN
Carriage_Return_Delay	0
Carriage_Return_Sequence	CR
Character_Flow_Control	ON
Code_Set	ASCII
Echoplex	OFF
End_Line_Character	CR
End_Line_Positioning	LFS
End_Output_Sequence	
End_Page_Action	NONE
End_Partial_Character	LF
End_Partial_Positioning	CRS
Fold_Line	OFF
Form_Feed_Delay	0
Form_Feed_Sequence	ESC [H ESC [J
Hold_Page	OFF
Hold_Page_Over	ON
Line_Feed_Delay	0
Line_Feed_Sequence	LF
Network_Command_Character	%
Page_Length	24
Page_Width	80
Parity	EVEN
Response_Action	SEND
Status_Action	SEND
Terminal_Model	MAC_CONNECT_xx[1]

1. With Control Data CONNECT version 1.0, set to 10; version 1.1, set to 11; version 2.0, set to 20; version 2.1, set to 21.

Table E-6. Selected Terminal Attribute Settings: DEC Terminals

Terminal Attribute	DEC VT100/VT220
Attention_Character	NUL
Backspace_Character	BS
Begin_Line_Character	NUL
Cancel_Line_Character	CAN
Carriage_Return_Delay	0
Carriage_Return_Sequence	CR
Character_Flow_Control	ON
Code_Set	ASCII
Echoplex	ON
End_Line_Character	CR
End_Line_Positioning	LFS
End_Output_Sequence	
End_Page_Action	NONE
End_Partial_Character	LF
End_Partial_Positioning	CRS
Fold_Line	ON
Form_Feed_Delay	0
Form_Feed_Sequence	ESC [H
Function_Key_Class	DEC_VTxxx[1]
	ESC [J
Hold_Page	OFF
Hold_Page_Over	OFF
Line_Feed_Delay	0
Line_Feed_Sequence	LF
Network_Command_Character	%
Page_Length	24
Page_Width	80
Parity	EVEN
Response_Action	SEND
Status_Action	SEND
Terminal_Model	DEC_VTxxx[2]

1. Attribute value for DEC VT100 is DEC_VT100. Attribute value for DEC VT220 is DEC_VT220.

2. Attribute value for DEC VT100 is DEC_VT100. Attribute value for DEC VT220 is DEC_VT110.

Table E-7. Selected Terminal Attribute Settings: Zenith Computers

Terminal Attribute	Zenith Z19/Z29
Attention_Character	NUL
Backspace_Character	BS
Begin_Line_Character	NUL
Cancel_Line_Character	CAN
Carriage_Return_Delay	0
Carriage_Return_Sequence	CR
Character_Flow_Control	ON
Code_Set	ASCII
Echoplex	OFF
End_Line_Character	CR
End_Line_Positioning	LFS
End_Output_Sequence	
End_Page_Action	NONE
End_Partial_Character	LF
End_Partial_Positioning	CRS
Fold_Line	OFF
Form_Feed_Delay	0
Form_Feed_Sequence	ESC [H ESC [J
Hold_Page	OFF
Hold_Page_Over	ON
Line_Feed_Delay	0
Line_Feed_Sequence	LF
Network_Command_Character	%
Page_Length	24
Page_Width	80
Parity	EVEN
Response_Action	SEND
Status_Action	SEND
Terminal_Model	ZEN_Zxx[1]

1. Attribute value for Zenith Z19 is ZEN_Z19. Attribute value for Zenith Z29 is ZEN_229.

3270 Binary Synchronous Communications Protocol F

3270 Binary Synchronous Communications Protocol F

The 3270 Binary Synchronous Communications protocol users can access NOS/VE and NOS services from an IBM 3270 Information Display System, connected via multipoint Binary Synchronous Communications (BSC) lines, with the BSC3270TIP terminal interface program (TIP).

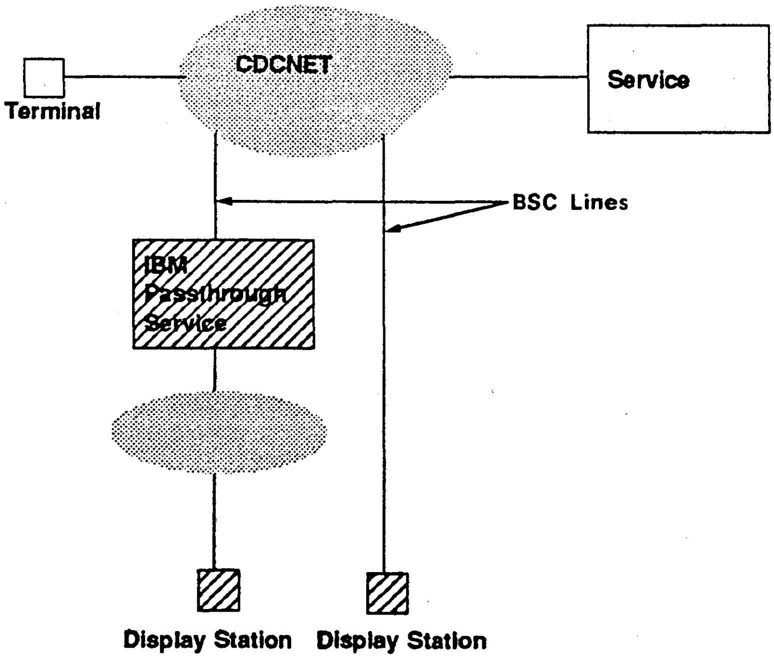
In general, this TIP follows the same conventions as those TIPs for other CDCNET-supported protocols. All commands, attributes, or other features supported by the 3270 Binary Synchronous protocol with similar requirements as those of the Asynchronous protocol are documented elsewhere in this manual.

This appendix supplements rather than replaces the other documentation. The following sections contain differences and unique capabilities that you may encounter while using your 3270 display station (terminal).

Section	Description
Your display station	Discusses your display station's place in the network and general screen capabilities.
Attribute support	Describes the subset of CDCNET terminal and connection attributes that the protocol supports.
Migrating from CCP	Highlights differences between CCP and CDCNET support.

Your Display Station

Your site administrator can configure a display station in two different ways depending upon the site's hardware and software requirements. Some display stations (and 3270 printers) connect directly to the network; others go through an IBM passthrough service before connecting to the network.



Capabilities

Since the network supports Nontransparent IBM Multipoint Bisynchronous Communications, you can use the CDCNET capabilities described in the rest of the manual when working at a 3270 display station connected to CDCNET. However, the screen display on your 3270 display station differs from that of other terminals. Unless you are using a full-screen application like Full Screen Editor (FSE), the network treats the display station like a "line-at-a-time" terminal.

- The first character of each line is reserved for the field attribute character. Therefore, the number of characters allowed on a line is usually one character less than the PAGE_WIDTH setting (that is, 79 out of 80 characters).

- When you enter input, it always appears after the last output line on the display. For example, if you enter the ASCII command (ascii), NOS responds with the command name (ASCII) and a prompt (/). Then you begin typing on the following line.

```
ascii
ASCII
/
(At this point, you enter input.)
```

Features Not Supported

Since CDCNET does not support Transparent IBM Multipoint Bisynchronous Communications, certain features normally available on 3270 display stations cannot be used. Unavailable features include color, extended highlighting, field validation, character attributes, extended character sets, and partitions. If your service attempts to use any of these features, it usually disconnects your display station. If this happens, call the site administrator and report the difficulty.

Attribute Support

You can change and display certain terminal and connection attributes from your 3270 display station.

Terminal Attributes

The 3270 Binary Synchronous Communications protocol supports a subset of the available terminal attributes. The BSC3270TIP supports this attribute subset. The default settings for these supported attributes are displayed in figure F-1.

```
Cancel_Line_Character      :NUL
Code_Set                   :EBCDIC
End_Page_Action            :NONE
Hold_Page                  :OFF
Hold_Page_Over             :OFF
Network_Command_Character  :%
Page_Length                :24
Page_Width                 :80
Response_Action            :SEND
Status_Action              :SEND
Terminal_Model             :IBM_3270
```

Figure F-1. 3270 Binary Synchronous Communications Protocol Terminal Attribute Defaults

Table F-1 compares the way 3270 uses certain terminal attributes with their quick-reference descriptions in chapter 3.

Table F-1. Terminal Attribute Differences

Terminal Attribute (Abbreviation)	Default Setting	Comparison
CANCEL_LINE_CHARACTER (CLC)	NUL	Same description applies in both cases. This attribute provides another way to erase, in addition to the following keys typically used at a 3270 display station. DEL ERASE EOF ERASE INPUT
CODE_SET (CS)	EBCDIC	CDCNET translates between ASCII and the code set used by 3270 equipment by default, which is EBCDIC. Site-defined code sets are also allowed.
END_PAGE_ACTION (EPA)	NONE	Same description applies in both cases.
HOLD_PAGE (HP)	OFF	Same description applies in both cases.
HOLD_PAGE_OVER (HPO)	OFF	Same description applies in both cases.

(Continued)

Table F-1. Terminal Attribute Differences *(Continued)*

Terminal Attribute (Abbreviation)	Default Setting	Comparison
NETWORK_ COMMAND_ CHARACTER (NCC)	%	Use any character in table B-6, except for data link control characters in table F-2. If you enter a NCC, press ENTER to go from the bottom of the page to the top when HOLD_PAGE is set to ON.
PAGE_LENGTH (PL)	24	You can use 12 through 43 lines. This is a subset of the values (0, 2 through 255) that the network provides.[1] The value 0, used in other protocols to suppress page division, is not valid for the 3270 protocol. To avoid display problems, specify your PAGE_LENGTH setting as the actual length of your screen. The following chart lists typical length and width combinations for screens on 3270 display stations.

Length (in Lines)	Width (in Characters)
24	40
24	80
27	132
32	80
43	80

1. If the page size (PAGE_LENGTH setting [lines] multiplied by PAGE_WIDTH [characters]) exceeds 3564, the result is unpredictable.

(Continued)

Table F-1. Terminal Attribute Differences *(Continued)*

Terminal Attribute (Abbreviation)	Default Setting	Comparison
PAGE_WIDTH (PW)	80	You can use 40 through 132 characters. This is a subset of the values (0, 10 through 255) that the network provides.[1] The value 0, used in other protocols to suppress line division, is not valid for the 3270 protocol. To avoid display problems, use the actual width of your screen as your PAGE_WIDTH setting. If your display station supports the ERASE_WRITE_ ALTERNATE command and has an alternate screen width, select the appropriate width. The following are alternate length (in lines) and width (in characters) combinations for screens on 3270 display stations. **Standard Size** / **Alternate Size** 32 by 80 / 27 by 132 43 by 80 / 27 by 132
RESPONSE_ACTION (RA)	SEND	Same description applies in both cases.
STATUS_ACTION (SA)	SEND	Same description applies in both cases.
TERMINAL_MODEL (TM)	IBM_3270	Same description applies in both cases.

1. If the page size (PAGE_LENGTH setting [lines] multiplied by PAGE_WIDTH [characters]) exceeds 3564, the result is unpredictable.

Data Link Control Characters

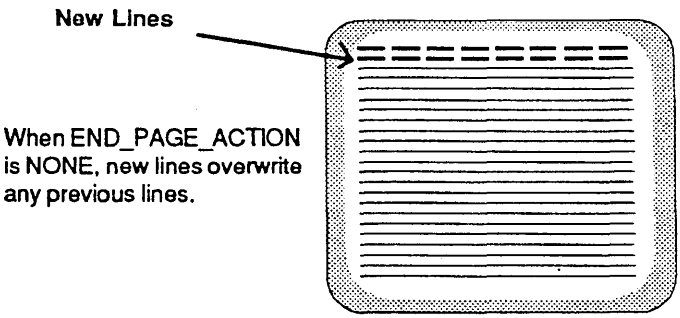
The data link control characters, listed in table F-2, control communications with your display station. Since they are reserved for this purpose, they may not appear in output from the service or in terminal attribute settings.

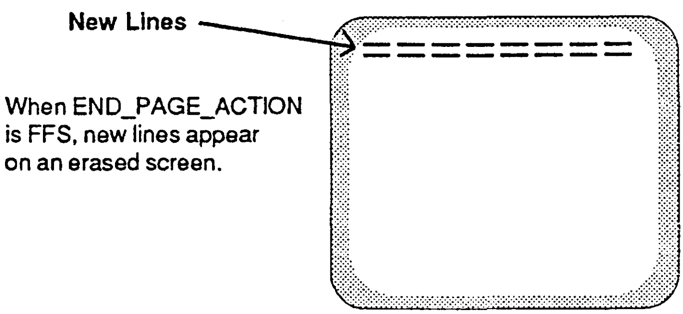
Table F-2. Data Link Control Characters

Graphic/Mnemonic	Name or Meaning
SOH	Start of heading
STX	Start of text
ETX	End of text
EOT	End of transmission
ENQ	Enquiry
ACK	Acknowledge
NAK	Negative acknowledge
SYN	Synchronous idle

Using END_PAGE_ACTION

If the network is formatting your screen in line mode, it repositions the cursor at the top of the screen after filling the last line with input or output. (This occurs when INPUT_EDITING_MODE is normal.) By setting END_PAGE_ACTION, you can control whether the network clears the screen when it moves the cursor.





Transparent Mode

When the service is sending transparent output and formatting your display, the network passes the data from the service to the display without editing it. (This occurs when INPUT_EDITING_MODE is transparent.) The service must then include all the necessary 3270 display orders to position the field attribute characters (not BSC characters) that format the screen.

Connection Attributes

The 3270 Binary Synchronous Communications protocol supports a subset of the available connection attributes. The BSC3270TIP supports this attribute subset. The default settings for these supported attributes are displayed in figure F-3.

```
Break_Key_Action                  : 1
Input_Block_Size                  : 160
Input_Editing_Mode                : NORMAL
Input_Output_Mode                 : UNSOLICITED
Transparent_Character_Mode        : TERMINATE
```

Figure F-2. 3270 Binary Synchronous Communications Protocol Connection Attribute Defaults

Table F-3 compares the way 3270 uses certain connection attributes with their quick-reference descriptions in chapter 4.

Table F-3. Connection Attribute Differences

Connection Attribute (Abbreviation)	Default Setting	Comparision
BREAK_KEY_ ACTION (BKA)	1	Same description applies in both cases.
INPUT_BLOCK_ SIZE (IBS)	160	Same description applies in both cases.
INPUT_EDITING_ MODE (IEM)	NORMAL	Same description applies in both cases.
INPUT_OUTPUT_ MODE (IOM)	UNSOLICITED	You can use all three settings. The display station only sends input when the network requests it, and the network sends output whenever the station is able to accept it. For unsolicited mode, the TIP immediately clears the next line after input and unlocks the keyboard before the input is forwarded to the host service. For solicited mode, the TIP leaves the keyboard locked after input until the host service solicits the next input. For either mode, the TIP clears the line following any output from the service and unlocks the keyboard.
TRANSPARENT_ CHARACTER_ MODE (TCM)	TERMINATE	Same description applies in both cases.

Migrating from CCP

If you have been using CCP and are migrating to CDCNET, your processing may be affected by some of the differences listed in table F-4. (Also see appendix D.)

Table F-4. CCP and CDCNET Differences

Feature	CDCNET Support	CCP Support
Secure input	Ensures a secure environment in which all existing connections are deleted before you establish one to your service. Performing the secure-access sequence prevents someone else from obtaining and using your validation information. 1. Press the PA1 key. 2. Wait for the service to reenable the keyboard. 3. Press A (lowercase or uppercase). 4. Press T (lowercase or uppercase). 5. Press the ENTER key.	Uses the following sequence: CLEAR (PA4) PA1 Network Security Character (CCP installation parameter)

(Continued)

Table F-4. CCP and CDCNET Differences *(Continued)*

Feature	CDCNET Support	CCP Support
Interrupts	Supports user interrupts. (See chapter 6 for more information.) You can issue an interrupt by doing either of the following: • Enter the NETWORK_ COMMAND_ CHARACTER followed by an integer or letter (for example, %1 and %2). • Press the PA1 key. It performs the interrupt specified by BREAK_KEY_ ACTION.	Provides interrupts called User Breaks. You can issue a User Break by doing either of the following: • Press the PA1 key for User Break 1. • Press the PA2 key for User Break 2.
Page holding	Stops output when HOLD_PAGE is ON at PAGE_LENGTH less one line. If the PAGE_ LENGTH setting is same as the actual screen length, your output stops at the end of the physical screen.	If the your setting is the same as the actual screen length, your output stops whenever the number of lines minus one has been displayed. If your output starts in the middle of the screen, it may stop in the middle of the screen.

(Continued)

Table F-4. CCP and CDCNET Differences *(Continued)*

Feature	CDCNET Support	CCP Support
Initiating transparent input	Establishes transparent input by setting INPUT_EDITING_MODE. Using this setting enables the service to control input.	Initiates transparent input by making the last output transmission transparent.
Terminating transparent input	Enables you to terminate transparent input (emergency transparent sequence) by escaping to $NET: 1. Press the PA1 key. 2. Wait for the service to reenable the keyboard. 3. Press A (lowercase or uppercase). 4. Press X (lowercase or uppercase). 5. Press the ENTER key.	Does not support a similar feature.

X.25 Asynchronous Protocol G

X.25 Asynchronous Protocol G

The X.25 asynchronous protocol allows asynchronous interactive terminal users to access a service across an X.25 link. CDCNET supports the X.25 asynchronous protocol through a terminal interface program (TIP). This TIP supports asynchronous terminals using the X.25 asychronous protocol that are connected to CDCNET through an X.29 PAD (Packet Assembler/Disassembler).

Features available to an X.25 asynchronous protocol user accessing CDCNET over an X.25 link are generally the same as the features available to users of asynchronous terminals connected directly to CDCNET.

Since CDCNET's implementation of X.25 follows the same conventions as the other CDCNET-supported protocols, this appendix supplements the existing CDCNET documentation. The following sections describe only the differences and unique capabilities of the terminal interface program supporting the X.25 asynchronous protocol. The commands, attributes, and features described in general CDCNET documentation also apply to connections made through an X.25 link, except where noted in this appendix.

Section	Description
The Protocol	Describes general features of the X.25 asynchronous protocol as supported by the X.25 asynchronous terminal interface program.
The Environment	Discusses the relationships among components in the connection from terminal to CDCNET.
Attribute Support and PAD Parameters	Describes the CDCNET terminal and connection attributes, and packet assembler/disassembler (PAD) parameter settings.

The Protocol

The X.25 asynchronous protocol, as supported by the X.25 asynchronous terminal interface program, provides asynchronous interactive terminal users access to CDCNET across an X.25 link. Recommendation X.25 (1980) is a data communications standard published by the Consultative Committee of International Telephone and Telegraph (CCITT). The terminal interface program conforms to this standard, which is part of the International Standard Organization/Open Systems Interconnection (ISO/OSI) model.

The Environment

The X.25 environment consists of:

- A terminal
- A packet assembler/disassembler (PAD)
- CDCNET, configured with the X.25_ASYNCTIP

Figure G-1 illustrates the relationships in this environment.

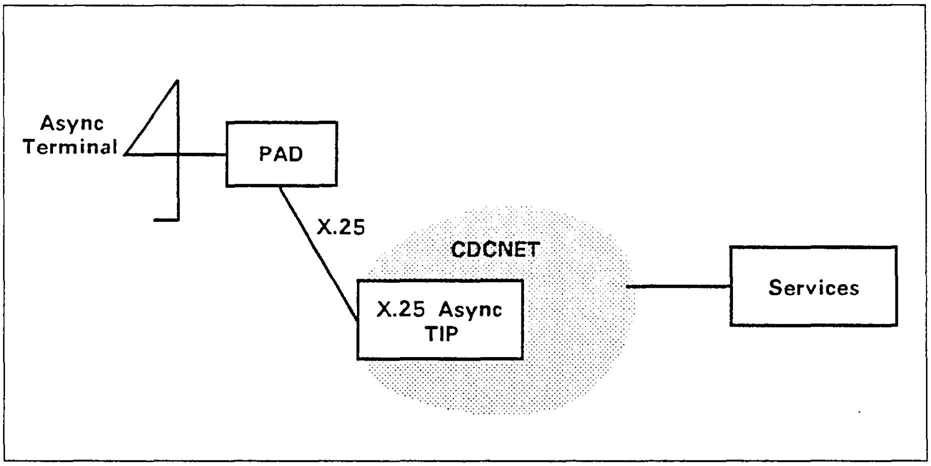


Figure G-1. X.25 Environment

The Terminal-to-PAD Connection

X.25 is a packet-switching protocol. So that terminals can communicate using this protocol, data from the terminal must be assembled into packets that conform to the protocol, and output to the terminal must be disassembled for use by the terminal. This function is performed by a packet assembler/disassembler, or PAD. A PAD can reside on an X.25 public data network (PDN), in specialized communications equipment, or is sometimes a feature of the workstation itself.

CDCNET and the X25_ASYNCTIP

CDCNET supports the X.25 protocol through its X.25 asynchronous terminal interface program, X25_ASYNCTIP. When CDCNET is configured to support the X.25 asynchronous protocol, the X25_ASYNCTIP resides in one or more of the device interfaces (DIs) on the network. The X25_ASYNCTIP works with the PAD to manage the terminal-to-application communications..

The X25_ASYNCTIP provides the following features:

- Performs validation of X.25 facilities in determining acceptance or rejection of an incoming call.
- Accesses and sets PAD parameters based on terminal and connection attributes specified by the application and terminal user.
- Performs character formatting/conversion between the virtual terminal protocol and the data sent to and from the terminal.

Using CDCNET across an X.25 link via the X25_ASYNCTIP is the same as using CDCNET across a direct connection on an asynchronous terminal via the ASYNCTIP, except for the following features:

- Automatic recognition
- User interrupt function
- BREAK key and attention character actions

These three exceptions are described on the following pages.

Automatic Recognition

When using the X.25 asynchronous protocol through the X25_ASYNCTIP, the automatic recognition sequence, if any, is performed by the PAD. CDCNET cannot control whether or how the PAD performs automatic recognition. CDCNET determines the speed of a remotely connected terminal by reading PAD reference #11 (see Attribute Support and PAD Parameters, later in this appendix). Use the CODE_SET terminal attribute to establish your terminal's code set (CHATA CS=value).

NOTE

The X25_ASYNCTIP does not recognize the ACTIVATE_AUTO_RECOGNITION (ACTAR) terminal user command.

User Interrupt Function

A user interrupt of 0 (%0) synchronizes the input path between CDCNET and the service or application. User interrupts in the range from 1 through 9 synchronize both the input and output paths between CDCNET and the service or application. User interrupts do not affect data that has already been passed from the X25_ASYNCTIP to the PAD. This data is still forwarded to the terminal or service.

BREAK Key and Attention Character Actions

Output data is lost if you strike the BREAK key while receiving output. When the PAD receives the BREAK key it stops delivering data to the terminal and discards all data that has been forwarded by the X25_ASYNCTIP.

The secure access sequence, described in chapter 6, can be used with the X25_ASYNCTIP to delete all existing connections.

You can terminate transparent input mode by entering the following sequence:

1. Press the BREAK key or enter the attention character.

2. Hold down the CONTROL (or CTRL) key and enter AX.

3. Press <CR>.

Only a BREAK key sequence is guaranteed to work regardless of the state of your virtual circuit.

Attribute Support and PAD Parameters

Connection and terminal attributes, as well as PAD parameters, affect the communications from terminal to application over an X.25 link.

Connection Attributes

The X25_ASYNCTIP supports all of the connection attributes available to asynchronous protocol users through the ASYNCTIP. The default values for all connection attributes supported by the X25_ASYNCTIP are the same as for the ASYNCTIP connection attribute values documented earlier in this manual.

NOTE

Input does not interrupt output while the value of the unsolicited INPUT_OUTPUT_MODE parameter is UNSOLICITED. The only way to interrupt output is by using the BREAK key or attention character, or X-OFF (only if the CHARACTER_FLOW_CONTROL terminal attribute is set to TRUE).

Output is deferred when the connection is in unsolicited mode and input is active. However, determination by the X.25 protocol of when input is active differs from a determination made by the asynchronous protocol. The X.25 asynchronous protocol considers input active when a data packet is received with the more-bit (M-bit) set, or one not terminated with a non-partial forwarding character (ELC/TFC/TTC). The asynchronous protocol considers input active when the first character of a message is entered.

Terminal Attributes

When you use CDCNET through an X.25 link, some of your terminal attributes are controlled directly by CDCNET, while others are controlled by the PAD, under the direction of CDCNET (CDCNET sets PAD references).

The CDCNET X25_ASYNCTIP manages the following terminal attributes directly:

Attention_Character (AC)
Backspace_Character (BC)
Begin_Line_Character (BLC)
Cancel_Line_Character (CLC)
Carriage_Return_Delay (CRD)
Carriage_Return_Sequence (CRS)
Code_Set (CS)
Control_Code_Replacement (CCR)
End_Line_Character (ELC)
End_Line_Positioning (ELP)
End_Output_Sequence (EOS)
End_Page_Action (EPA)
End_Partial_Character (EPC) [1]
End_Partial_Positioning (EPP)
Fold_Line (FL)
Form_Feed_Delay (FFD)
Form_Feed_Sequence (FFS)
Hold_Page (HP)
Hold_Page_Over (HPO)
Line_Feed_Delay (LFD)
Line_Feed_Sequence (LFS)
Network_Command_Character (NCC)
Page_Length (PL)
Page_Width (PW)
Parity (P)
Response_Action (RA) [2]
Status_Action (SA)
Terminal_Model (TM)

---

1. Performance is optimal if the EPC and EPP attributes are set to their negative values (EPC=NULL, EPP=NONE).

2. The RA terminal attribute has no affect on any PAD parameter settings.

The following terminal attributes are set in the PAD by CDCNET, but are controlled and used by the PAD.

Terminal Attribute	Description
Character_Flow_Control (CFC)	Flow control is managed by the PAD, not by CDCNET. CDCNET can manipulate flow control, however, by setting the appropriate PAD parameters. EIA flow control using RTS and CTS is not supported by the X25_ASYNCTIP.
Echoplex (E)	CDCNET does not echo received input. If echoplex is turned on and data is typed ahead while the connection is in solicited mode, the data is echoed immediately by the PAD, not when processed by CDCNET. In certain environments, the PAD sends line feeds if the terminal attribute echoplex (E) is set to TRUE. Assuming the default values of ELC=CR, ELP=LFS, and LFS=LF, CDCNET need not send END_LINE_POSITIONING (ELP).
Parity (P)	CDCNET generates the correct parity for output to the terminal. Checking for correct parity on input is a function of the PAD.

PAD Parameter Settings

If you do not change the CDCNET terminal and connection attributes, the X25_ASYNCTIP sets PAD parameters to their default values at initial connection time (consult CCITT Recommendation X.3). These default values, which are listed in the following table, can be adjusted using the CHACA and CHATA commands.

Table G-1. PAD Parameter Values

PAD Parameter	Description	Default Value	Optional Values
1	PAD recall using a character	1(10)	If the input editing mode (IEM) is transparent, then this parameter is set to zero (0).
2	Echoplex	1(10)	If the echoplex terminal attribute is not TRUE, this PAD parameter is set to zero (0).
3[1]	Selection of data-forwarding signal	34(10)	If IEM=NORMAL, this parameter is ELC + AC (if AC≠NULL) + EPC (if EPP=T). If input editing mode is transparent, this parameter is the aggregate of AC and the type of transparent mode, as follows: • If TTM or TLM, then zero (0). • If TCM=F, then TFC.

1. CDCNET defines the transparent forwarding and terminating characters (TFC/TTC) as 8-bit characters, but since the X.25 specification has no provision for mapping 8-bit characters to PAD parameter 3, the X25_ASYNCTIP does not attempt to map these characters to parameter 3 or 4 if the higher order bit is set.

(Continued)

Table G-1. PAD Parameter Values *(Continued)*

PAD Parameter	Description	Default Value	Optional Values
3[1] (Continued)			• If TCM=T, then TTC. • If TCM=FT, then TFC and TTC. • If no transparent mode is selected, then zero (0).
4[2]	Selection of idle timer delay (in 0.05 second units)	0(10)	If input editing mode is transparent and: • TTM is selected, then 1(10) through 20(10) based on PAD parameter 11. • TLM is selected, then 20(10). If TML=1, then 1(10). • No transparent mode is selected, then 20(10). If input editing mode is normal, then 20(10).
5	Ancillary device control	0(10)	If CFC=TRUE, then 1.

1. CDCNET defines the transparent forwarding and terminating characters (TFC/TTC) as 8-bit characters, but since the X.25 specification has no provision for mapping 8-bit characters to PAD parameter 3, the X25_ASYNCTIP does not attempt to map these characters to parameter 3 or 4 if the higher order bit is set.

2. If an AC, ELC, EPC, TTC, TFC character cannot be mapped to parameter 3, or if the computed value for parameter 3 is rejected by the PAD (nonsupported value), parameter 4 is set to 20(10).

(Continued)

Table G-1. PAD Parameter Values *(Continued)*

PAD Parameter	Description	Default Value	Optional Values
6	Control of PAD service signals	N/A	This PAD parameter is not modified or referenced by CDCNET.
7	Selection of operation of PAD on receipt of break signal from the start-stop mode DTE	21(10)	N/A
8	Discard output	0(10)	N/A
9	Padding after carriage return (CR)		This PAD parameter is not modified or referenced by CDCNET.
10	Line folding	0(10)	N/A
11	Binary speed of start-stop mode DTE	N/A	This is a read-only parameter that is never modified. It is referenced when computing FFD, CRD and LFD NULs.

(Continued)

Table G-1. PAD Parameter Values *(Continued)*

PAD Parameter	Description	Default Value	Optional Values
12[1]	Flow control of the PAD by the start-stop mode DTE	0(10)	If CFC=TRUE, then 1(10).
13	Linefeed insertion after carriage return	4(10)	If IEM=NORMAL, E=TRUE, ELC=CR, and ELP=LF, then 4; if EPC=CR, and EPP=LF, then 4; otherwise, zero.
14[2]	Padding after linefeed		Not applicable.
15	Editing	0(0)	Not applicable.
16[2]	Character delete		Not applicable.
17[2]	Line delete		Not applicable.
18[2]	Line display		Not applicable.
19[2]	Editing PAD service signals		Not applicable.
20[2]	Echo mask		Not applicable.
21	Parity treatment	0(10)	Not applicable.
22[1]	Page wait	0(10)	Not applicable.

1. PAD parameters 12 through 22 provide user facilities that are not necessarily provided in all PADs.

2. PAD parameters 14, and 16 through 20 are not modified or referenced by CDCNET.

If an error PAD message is received in response to a setting of parameter 3, CDCNET sets parameter 3 to 126 (this means that all control characters, and the DEL character, are used as data-forwarding signals; in other words, single-character forwarding is used). Any other errors reported by the PAD are ignored.

CDCNET recomputes the values of the PAD parameters each time a CDCNET attribute is changed (by a terminal user or application, or when terminating transparent mode). If the computed values are different (previously computed values are maintained for each virtual circuit), a set pad parameter message is sent to the PDN PAD with the updated values.

X.PC Communications Protocol H

X.PC Communications Protocol H

X.PC[1] is an asynchronous data communications protocol developed to improve the networking capabilities of microcomputers. CDCNET support of the X.PC protocol and the microcomputer X.PC data communications software provides powerful capabilities such as windowing, delivery-assurance of all data transfers, and multiple, simultaneous connections between separate applications and microcomputer windows over low-cost communication lines.

Since the implementation of X.PC on CDCNET follows the same conventions as do the other CDCNET-supported protocols, this appendix supplements the existing CDCNET and microcomputer documentation. The following sections describe the differences and unique capabilities of the software supporting the X.PC protocol. The commands, attributes, and features in the general CDCNET documentation also apply to connections made through an X.PC link, except where noted in this appendix.

Section	Description
The Protocol	Describes general features of the X.PC protocol as supported by CDCNET.
The Environment	Discusses your microcomputer requirements, the CDCNET network, and X.PC operational characteristics.
Attribute Support	Describes the CDCNET terminal and connection attributes which are supported by the X.PC protocol.

1. X.PC is a trademark of TYMNET, Incorporated.

The Protocol

The X.PC protocol program provides the user of a microcomputer with enhanced access to CDCNET through the use of packet-switching data communications techniques. X.PC was derived from the Consultative Committee of International Telephone and Telegraph Recommendation X.25 (CCITT X.25 or X.25) and conforms to the International Standard Organization/Open Systems Interconnection (ISO/OSI) model. The name X.PC was chosen to indicate the functional similarity to the X.25 recommendation. The CDCNET support of the protocol conforms to this standard.

X.PC is an error-checking asynchronous protocol with the ability to support up to 15 simultaneous logical channels over a single physical communications link. Error checking and recovery capability is provided by using sequenced packets and CCITT 16-bit cyclic redundancy checksum (CRC-16). A logical channel identifier that provides for multiple logical connections is built into the packet header.

The Environment

The X.PC environment consists of:

- A microcomputer
- X.PC data communications software for the microcomputer
- CDCNET, configured for X.PC support
- Up to 15 separate host or information services

Figure H-1 illustrates the X.PC environment

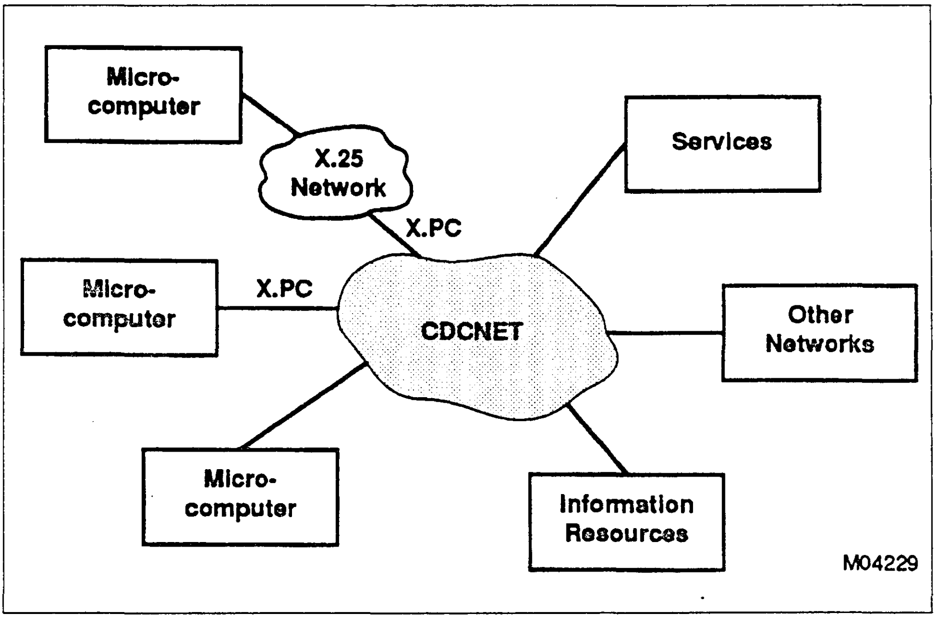


Figure H-1. X.PC Environment

The Microcomputer Connection

The microcomputer is connected to CDCNET through one of the following:

- A full-duplex, RS-232, asynchronous interface, with or without modems. The line can be dedicated or switched. The carrier must be constant. Line speeds are from 75 to 38,400 bits per second (bps). These lines are configured with the Asynchronous or X.PC protocols.
- An X.25 packet-switching network configured to support the X.25 Asynchronous protocol.

Automatic recognition of line speed is allowed only for the first type of connection.(Refer to the ACTIVATE_AUTO_RECOGNITION command in chapter 2). The X.PC character set is always ASCII, and parity is set to zero or none by CDCNET. See the terminal attribute PARITY description in table H-1.

NOTE

Certain microcomputer X.PC software products support only line speeds equal to, or greater than 600 bps. Reference the specific documentation for your microcomputer data communications software to determine whether this line speed requirement is applicable to your system.

The Microcomputer's Data Communications Software

The X.PC protocol is implemented in two parts: an X.PC driver within your microcomputer's data communications software; and a terminal interface program (TIP), which resides within CDCNET. The microcomputer's X.PC driver is used by microcomputer applications to build and transmit (or to receive and disassemble) packets of information which are exchanged between the microcomputer and CDCNET. The microcomputer's X.PC driver software provides access to the specific features of X.PC on CDCNET.

Control Data's Desktop/VE, Microsoft's ACCESS, and Microstuff's CROSSTALK[2] are examples of microcomputer software products which support the X.PC protocol. The documentation for these products, or any other microcomputer X.PC data communications products, should provide instructions for installing and initializing X.PC on your microcomputer. Reference the documentation for your microcomputer software to determine the explicit X.PC protocol features your microcomputer supports.

NOTE

The CDCNET implementation of the X.PC protocol supports microcomputer data communications software which has passed TYMNET certification. To avoid problems, please ensure that your microcomputer software has this certification. This can be accomplished by referencing the documentation with your microcomputer data communications software or by asking the vendor.

CDCNET and the XPC Protocol

The other part of the X.PC protocol implementation is the software within CDCNET. The software differs depending on how the microcomputer is connected to CDCNET.

2. ACCESS and CROSSTALK are registered trademarks.

CDCNET Configuration of X.PC on an Asynchronous Line

The CDCNET site administrator can configure an X.PC or an asynchronous data communications line in two different ways. Some lines can be connected directly to the XPCTIP; others can be initially connected to the ASYNCTIP as an asynchronous ASCII terminal before being switched to the XPCTIP. The second of these options provides greater flexibility in allowing microcomputers and standard data communications terminals to connect to the same switched lines. Contact your CDCNET site administrator to determine which configuration is available to your microcomputer.

If your microcomputer is not initially connected to the XPCTIP, you can switch to the X.PC protocol by issuing the terminal user command ACTIVATE_X_PERSONAL_COMPUTER (ACTXPC). ACTXPC can be issued directly by the microcomputer user or from within the microcomputer's data communications software. ACTXPC is described in chapter 2 of this manual; however, you should reference the documentation for your microcomputer's data communications software to understand how the X.PC connection is accomplished on your system.

The CDCNET site administrator can configure the X.25_ASYNCTIP for microcomputers connected via an X.29 PAD facility. No special configuration is required for X.PC support. CDCNET does no switching between terminal interface programs when X.PC is run over this type of interface. A terminal user activates the X.PC protocol by using the terminal user command ACTXPC.

NOTE

For lines using the X.PC protocol, user connections are counted in the following manner: each logical channel without any service connections is counted as having one user connection. The number of user connections from any other logical channel is equal to the number of service connections it has.

When the user connection limit is reached on a line using the X.PC protocol, no new logical channels are permitted, and existing logical channels with active service connections are not permitted to create new service connections. Logical channels without any existing service connections are permitted to create a single service connection.

Attribute Support

You can change and display certain terminal and connection attributes from your X.PC microcomputer. However, some microcomputer data communications software products can set these terminal and connection attributes directly. Reference the documentation with your microcomputer software to determine whether you should change any of these attributes with CDCNET commands.

Terminal Attributes

CDCNET supports a set of terminal attributes that apply to all of your terminal's X.PC connections. The default settings of these terminal attributes are displayed in figure H-2.

```
Attention_Character          :NUL
Backspace_Character          :BS
Cancel_Line_Character        :CAN
Carriage_Return_Delay        :0
Carriage_Return_Sequence     :CR
Code_Set                     :ASCII
Echoplex                     :OFF
End_Line_Character           :CR
End_Line_Positioning         :LF
End_Output_Sequence          :¹
End_Page_Action              :NONE
End_Partial_Character        :LF
End_Partial_Positioning      :CR
Fold_Line                    :ON
Form_Feed_Delay              :0
Form_Feed_Sequence           :FF
Hold_Page                    :OFF
Hold_Page_Over               :ON
Line_Feed_Delay              :0
Line_Feed_Sequence           :LF
Network_Command_Character    :%
Page_Length                  :24
Page_Width                   :80
Parity                       :NONE
Response_Action              :SEND
Status_Action                :SEND
Terminal_Model               :NONE
Backspace_Window             :PW

¹The default EOS is an empty sequence.
```

Figure H-2. X.PC Protocol Terminal Attribute Defaults

The following list of terminal attributes are only supported as shown in figure H-2, and cannot be changed:

- Backspace_Character
- Carriage_Return_Delay
- Code_Set
- Echoplex
- End_Line_Character
- End_Partial_Character
- Form_Feed_Delay
- Line_Feed_Delay

For those terminal attributes that the X.PC protocol treats differently from the asynchronous protocol, table H-1 contains the default setting and a comparison of the way X.PC protocol uses the attribute relative to the quick-reference description in chapter 3.

Some of the terminal attributes are supported with a fixed value. If changed by the terminal user or microcomputer data communications software, the change is rejected or ignored.

Table H-1. Terminal Attribute Differences

Terminal Attribute (Abbreviation)	Default Setting	Chapter 3 Comparison
ATTENTION_CHARACTER (AC)	NUL	Same description applies.
BACKSPACE_CHARACTER (BC)	BS	Cannot be changed.
BEGIN_LINE_CHARACTER (BLC)		Unused.
CANCEL_LINE_CHARACTER (CLC)	CAN	Same description applies. This option can be deactivated by assigning NUL to this attribute.
CARRIAGE_RETURN_DELAY (CRD)	0	Cannot be changed.

(Continued)

Table H-1. Terminal Attribute Differences *(Continued)*

Terminal Attribute (Abbreviation)	Default Setting	Chapter 3 Comparison
CARRIAGE_RETURN_SEQUENCE (CRS)	CR	Same description applies.
CHARACTER_FLOW_CONTROL (CFC)		Unused.
CODE_SET (CS)	ASCII	Cannot be changed.
ECHOPLEX (E)	OFF	Cannot be changed.
END_LINE_CHARACTER (ELC)	CR	Cannot be changed.
END_LINE_POSITIONING (ELP)	LF	Same description applies. The allowed values are: NONE, CR, LF, CRLF.
END_OUTPUT_SEQUENCE (EOS)	nothing sent	Same description applies. Defaults to send no extra characters.
END_PAGE_ACTION (EPA)	NONE	Same description applies.
END_PARTIAL_CHARACTER (EPC)	LF	Cannot be changed.
END_PARTIAL_POSITIONING (EPP)	CR	Same description applies. Allowed values are: NONE, CR, LF, CRLF.
FOLD_LINE (FL)	ON	Same description applies.
FORM_FEED_DELAY (FFD)	0	Cannot be changed.
FORM_FEED_SEQUENCE (FFS)	FF	Same description applies.

(Continued)

Table H-1. Terminal Attribute Differences *(Continued)*

Terminal Attribute (Abbreviation)	Default Setting	Chapter 3 Comparison
HOLD_PAGE (HP)	OFF	Same description applies.
HOLD_PAGE_OVER (HPO)	ON	Same description applies.
LINE_FEED_DELAY (LFD)	0	Cannot be changed.
LINE_FEED_SEQUENCE (LFS)	LF	Same description applies.
NETWORK_COMMAND_ CHRACTER (NCC)	%	Same description applies.
PAGE_LENGTH (PL)	24	Same description applies.
PAGE_WIDTH (PW)	80	Same description applies.
PARITY (P)	NONE	Allowed values are ZERO and NONE. Parity type of NONE has significance only during transparent input or output. In all other cases, parity is treated as ZERO.
RESPONSE_ACTION (RA)	SEND	Same description applies.
STATUS_ACTION (SA)	SEND	Same description applies.
TERMINAL_MODEL (TM)	NONE	Same description applies.

Connection Attributes

The X.PC protocol supports a set of connection attributes that regulate certain aspects of the connection between your microcomputer and the CDCNET service. The default settings for these connection attributes are displayed in figure H-3.

```
Attention_Character_Action        : 2
Break_Key_Action                  : 0
Input_Block_Size                  : 160
Input_Editing_Mode                : NORMAL
Input_Output_Mode                 : UNSOLICITED
Partial_Character_Forwarding      : OFF
Store_Backspace_Character         : OFF
Store_Nuls_Dels                   : OFF
Transparent_Character_Mode        : TERMINATE
Transparent_Forward_Character     : CR
Transparent_Length_Mode           : NONE
Transparent_Message_Length        : 255
Transparent_Terminate_Character   : CR
```

Figure H-3. X.PC Protocol Connection Attribute Defaults

For each connection attribute that the X.PC protocol treats differently from the asynchronous protocol, table H-2 contains the default setting and a comparison of the way X.PC protocol uses the attribute relative to the quick-reference description in chapter 4.

Table H-2. Connection Attribute Differences

Connection Attribute (Abbreviation)	Default Setting	Chapter 4 Comparison
ATTENTION_CHARACTER_ACTION (ACA)	2	Same description applies. Allowed values are 0 through 9. Values 1 through 9 cancel output as well as input. Depending upon the buffering and the actions taken locally at your microcomputer, there can be a delay before output stops.
BREAK_KEY_ACTION (BKA)	0	Same description applies. Allowed values are 0 through 9. Depending upon the buffering and the actions taken locally at your microcomputer, there can be a delay before output stops.

(Continued)

Table H-2. Connection Attribute Differences *(Continued)*

Connection Attribute (Abbreviation)	Default Setting	Chapter 4 Comparison
INPUT_BLOCK_SIZE (IBS)	160	Same description applies.
INPUT_EDITING_MODE (IEM)	NORMAL	Same description applies.
INPUT_OUTPUT_MODE (IOM)	UNSOLICITED	Same description applies.
PARTIAL_CHARACTER_FORWARDING (PCF)	OFF	Same description applies.
STORE_BACKSPACE_CHARACTER (SBC)	OFF	Cannot be changed.
STORE_NULS_DELS (SND)	OFF	Cannot be changed.
TRANSPARENT_CHARACTER_MODE (TCM)	TERMINATE	Same description applies.
TRANSPARENT_FORWARD_CHARACTER (TFC)	CR	Same description applies.
TRANSPARENT_LENGTH_MODE (TLM)	NONE	Same description applies.
TRANSPARENT_MESSAGE_LENGTH (TML)	255	Same description applies.
TRANSPARENT_TERMINATE_CHARACTER (TTC)	CR	Same description applies.
TRANSPARENT_TIMEOUT_MODE (TTM)		Unused.

Mode 4 Protocol I

Mode 4 Protocol I

The Mode 4 protocol is a data communications protocol supporting two-way alternate communcations (messages may be sent in one direction or another, but not in both directions simultaneously). The Mode 4 terminal interface program (TIP) enables synchronous terminals using the Mode 4 protocol to communicate with NOS and NOS/VE hosts using CDCNET. In general, this protocol adheres to the same conventions as other CDCNET-supported protocols.

Since CDCNET's implementation of the Mode 4 protocol follows the same conventions as other CDCNET-supported protocols, this appendix supplements, rather than replaces, the other documentation. The following sections descibe differences and unique capabilities that you may encounter while using your Mode 4 terminal. The commands, attributes, and features described in general CDCNET documentation also apply to connections made using the Mode 4 protocol, except where noted in this appendix.

Topic	Description
Your Mode 4 Terminal	Discusses Mode 4 terminals.
Supported Features	Lists the features of the Mode 4 protocol as supported by the Mode 4 terminal interface program.
Attribute support	Describes the subset of CDCNET terminal and connection attributes supported by the asynchronous protocol that are supported by the Mode 4 protocol.
Migrating from CCP	Highlights differences between CCP and CDCNET support.

Your Mode 4 Terminal

Two types of Mode 4 terminals are supported by the Mode 4 protocol as implemented by the MODE4TIP:

- Mode 4A terminals
- Mode 4C terminals

Mode 4A Terminals

Mode 4A terminals can consist of a group of several clusters consisting of one console, a line printer, and a card reader. Each terminal cluster must have one console device, and can have one optional card reader and one optional line printer.

Mode 4A console operation is suspended when either the card reader or the line printer becomes active.

Mode 4A terminals include the CDC 200UT, CDC 731, CDC 732, CDC 734, and equivalent terminals on a cluster.

Mode 4C Terminals

Mode 4C terminals can consist of several clusters of consoles and line printers. Each terminal cluster must have at least one console and can have up to 14 additional optional devices. These optional devices can be either consoles or line printers.

All Mode 4C devices can be active at a given time. Interactive operations can be concurrent with batch operations.

Mode 4C terminals include the CDC 711, CDC 714, and equivalent terminals.

Supported Features

The CDCNET MODE4TIP supports the following features:

- Synchronous line speeds from 1200 through 19200 bits per second
- Multiple terminal clusters on the same communications line
- Automatic recognition of code set, ASCII or BCD
- Virtual terminal protocol (VTP) for console devices
- Normal and transparent input/output mode for console devices
- /*STAB (START_BATCH) command for switching Mode 4A terminals from interactive mode to batch mode
- Status request and device polling. Cluster polling *is not* supported

Multiple Connections

The MODE4TIP supports multiple connections for one console. See chapter 1 for information about managing these connections.

Paging

Output sent to the console can be displayed page by page. When you enable the HOLD_PAGE terminal attribute, the MODE4TIP stops output to the terminal after PAGE_LENGTH minus 1 lines are displayed. When the HOLD_PAGE_OVER terminal attribute is also enabled, the TIP stops output after PAGE_LENGTH minus 2 lines are output and displays an <OVER> message at the bottom of the screen.

Output resumes when you enter any input line. If you enter an empty input line or the NETWORK_COMMAND_CHARACTER (%) followed by SEND, no input data is sent upline and the next page is displayed.

NOTE

For Mode 4A equipment that cannot recognize an empty line, use the NETWORK_COMMAND_CHARACTER (%) to page forward.

If the PAGE_WIDTH terminal attribute value is set to 0, the paging options are ignored and no paging takes place. This also eliminates normal cursor positioning to the beginning of the next line after each input or output block.

Interactive Interrupt

You can switch your Mode 4A terminal from interactive mode to batch mode by entering the /*STAB command at the console.

The MODE4TIP may also switch from interactive to batch mode if a file transfer to the printer begins and no input or output is received or sent to the console for 20 seconds.

You must enter the /*STAB command to begin reading cards unless the terminal is already in batch mode because the printer is active.

If batch input or output is interrupted by pressing the batch interrupt key/switch or because output data was received for the console, you may either wait 20 seconds for batch output to resume, or you may enter a /*STAB command to resume batch mode.

If batch input or output is interrupted because a printer message (PM) was encountered in the print file and sent to the console, the operator must first acknowledge the message on an OPES or RBF command before batch input/output can resume.

When batch input/output completes, the MODE4TIP automatically resumes polling for interactive input.

Automatic Recognition of Code Set

The MODE4TIP performs automatic recognition of code set when a communication line becomes active and the terminal type is not known to be either Mode 4A or Mode 4C. If the terminal type is known to be Mode 4C from the configuration, the code set is assumed to be ASCII and code set recognition is completed. If the terminal type is either unknown or known to be 4A from the configuration, the Mode 4 TIP waits for the first SEND or ETX to be entered.

Automatic recognition must complete before the MODE4TIP knows the code set. While the MODE4TIP is waiting for automatic recognition input to be entered, no devices on the cluster are active.

After logging out, 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 start new host connection and login procedures without repeating any of the preliminary dialup or automatic recognition access procedures.

Security Character Sequence

The MODE4TIP recognizes a special 4-character sequence of /*AT (the AT must be uppercase). When you enter this character sequence on a Mode 4 console as the first and only four characters of a line, all interactive connections for the console are disconnected.

Any files being transferred are completed, but new files are not started. The terminal can be disconnected from the network after all devices are idle.

Virtual Line Mode

Virtual Line Mode output consists of ASCII text. Each output line can be preceded by an optional ASCII character identified as a format effector. Output lines longer that the set value of PAGE_WIDTH are continued on the next line.

The format effector at the beginning of each line specifies vertical spacing and is optional. If a format effector is present, the MODE4TIP converts the format effector to the display operations shown in table I-1.

Table I-1. Mode 4 Console Format Effectors

Format Effector	Before Output [1]	After Output [1]
Space	None	Space 1 line
0 (zero)	Space 1 line	Space 1 line
- (dash)	Space 2 lines	Space 1 line
+ (plus)	None	Space 1 line
1	Clear screen and move cursor to top of screen	Space 1 line
*	Move cursor to top of screen	Space 1 line

1. After each input and output line, the TIP returns the cursor to the beginning (left) margin of the next line.

All other format effectors are treated as a space.

If PAGE_WIDTH=0, the MODE4TIP does not move the cursor to the beginning of the next line after input or output; this allows the application to control the cursor in Virtual Line Mode.

Transparent Input Mode

When transparent input editing mode is specified, the MODE4TIP removes the Mode 4 protocol header and trailer characters, but no code translation takes place. Cursor positioning to the next line after input or output is also eliminated.

The MODE4TIP supports the TRANSPARENT_CHARACTER_MODE (TCM) terminal attribute. TRANSPARENT_LENGTH_MODE (TLM) and TRANSPARENT_TIMEOUT_MODE (TTM) are not supported.

Transparent Character Mode is supported as follows:

TCM Value	Description
FORWARD	Each protocol block received from the device is forwarded as nonpartial data and transparent input mode remains in effect.
FORWARD and TERMINATE	The next protocol block received from the device is forwarded as nonpartial data and transparent input mode remains in effect.
NONE	Each protocol block received from the device is forwarded as partial data and transparent input mode is in effect.
TERMINATE	The next protocol block received from the device is forwarded as nonpartial data and transparent input mode does not remain in effect.

Transparent Output Mode

Selection of output type is controlled by the application through the selection of the data attributes included for each downline data block.

In transparent output mode, the MODE4TIP outputs the data to the device as received. Parity is added to each character. Mode 4 protocol header and trailer characters are also added. No code translation or cursor positioning is performed.

The application program sending the data is responsible for ensuring that the transparent data is acceptable to the terminal.

CDC 714-30 Terminal Underline Character

The CDC 714-30 Mode 4C terminal does not have an underline (_) character on its keyboard. Since the _ character is frequently used in NOS/VE SCL names, a special feature of the MODE4TIP allows you to use the tilde (~) character in place of the _ character. The ~ character is converted by the MODE4TIP to an _ character for both input from and output to the terminal. In this way, a CDC 714-30 terminal can be used on a NOS/VE system.

To enable this feature, use the CHANGE_TERMINAL_ATTRIBUTE (CHATA) command with the TERMINAL_MODEL (TM) parameter to change your terminal model name to MODE4C30:

```
%CHATA TM=MODE4C30
```

This command can be entered on your terminal or placed in a TUP file that is called whenever the line becomes active.

When this feature is enabled and the NETWORK_COMMAND_CHARACTER is 7E(16) (the ~ character), the NETWORK_COMMAND_CHARACTER is stored internally as the underscore character (_). This allows you to enter the ~ character as the NETWORK_COMMAND_CHARACTER. (This is necessary because, on input, the _character is substituted for a ~ character before any testing of the input character against the NCC is performed.)

To disable this feature, enter the CHANGE_TERMINAL_ATTRIBUTE command with the TERMINAL_MODEL parameter specified as any character string other than MODE4C30. (The default is MODE4.)

Attribute Support

You can change and display certain terminal and connection attributes from your Mode 4 terminal.

Terminal Attributes

The Mode 4 protocol, as supported by the MODE4TIP uses the default terminal attributes displayed in figure I-1.

```
Cancel_Line_Character       :NUL
End_Output_Sequence         :NONE
Fold_Line                   :TRUE
Hold_Page                   :OFF
Hold_Page_Over              :ON
Network_Command_Character   :%
Page_Length                 :24
Page_Width                  :80
Response_Action             :SEND
Status_Action               :SEND
Terminal_Model              :MODE4
```

Figure I-1. Mode 4 Protocol Terminal Attribute Defaults

For terminal attributes that the Mode 4 protocol treats differently from the Asynchronous protocol, table I-2 compares the way the MODE4TIP uses these terminal attributes with the way ASYNCTIP uses the attributes, as described in chapter 3.

Table I-2. Terminal Attribute Differences

Terminal Attribute (Abbreviation)	Default Setting	Chapter 3 Comparison
CANCEL_LINE_CHARACTER (CLC)	NUL	Same description applies in both cases.
END_OUTPUT_SEQUENCE (EOS)	None	Same description applies in both cases.
HOLD_PAGE (HP)	OFF	Same description applies in both cases.
HOLD_PAGE_OVER (HPO)	ON	Same description applies in both cases.
NETWORK_COMMAND_CHARACTER (NCC)	%	Same description applies in both cases.
PAGE_LENGTH (PL)	24	Same description applies in both cases.
PAGE_WIDTH (PW)	80	Same description applies in both cases.
RESPONSE_ACTION (RA)	SEND	Same description applies in both cases.
STATUS_ACTION (SA)	SEND	Same description applies in both cases.
TERMINAL_MODEL (TM)	MODE4	Same description applies in both cases.

Connection Attributes

Figure I-2 displays the default connection attributes supported by the Mode 4 protocol via the MODE4TIP.

```
Input_Block_Size                  : 160
Input_Editing_Mode                : NORMAL
Transparent_Character_Mode        : TERMINATE
```

Figure I-2. Mode 4 Protocol Connection Attribute Defaults

For connection attributes that the Mode 4 protocol treats differently from the Asynchronous protocol, table I-3 compares the way the MODE4TIP uses these connection attributes with the way ASYNCTIP uses the attributes as described in chapter 4.

Table I-3. Connection Attribute Differences

Connection Attribute (Abbreviation)	Default Setting	Chapter 4 Comparison
INPUT_BLOCK_SIZE (IBS)	160	Same description applies in both cases.
INPUT_EDITING_MODE (IEM)	NORMAL	7-bit ASCII, user interrupt, transparent mode, and the CANCEL_LINE_CHARACTER are supported. The BREAK key, the ATTENTION_CHARACTER, X-ON/X-OFF flow control, PARTIAL_CHARACTER_FORWARDING, and the BACKSPACE_CHARACTER are not supported.
TRANSPARENT_CHARACTER_MODE (TCM)	TERMINATE	See Transparent Input Mode in this appendix for a discussion of Mode 4 support of the TRANSPARENT_CHARACTER_MODE connection attribute.

Migrating from CCP

If you have been using CCP and are migrating to CDCNET, your processing may be affected by some of the differences listed in table I-4. (Also see appendix D.)

Table I-4. CCP and CDCNET Differences

Feature	CDCNET Support	CCP Support
Automatic recognition	Only the code set that your terminal uses is automatically recognized.	CCP automatically recognizes the cluster address and device configuration also.
Bottom-of-form	Supports skipping to the bottom-of-form (BOF) format effector on impact printers.	Does not support a similar feature.
Page width	If PW=0, then CDCNET does not position the cursor to the beginning of the next line after input or output.	Does not support a similar feature.

(Continued)

Table I-4. CCP and CDCNET Differences *(Continued)*

Feature	CDCNET Support	CCP Support
Resume batch mode	Uses the /*STAB command to switch from interactive mode to batch mode.	Uses the Remote Batch Facility GO or RESUME command. The GO or RESUME command causes the terminal to return immediately to batch mode if at least one batch input or output device is active.
Secure input	Clear the screen and type, in uppercase only: /*AT (send key).	Clear the screen and type the network security character (CCP installation parameter).

3270 SNA Communications Protocol J

3270 SNA Communications Protocol J

The 3270 Systems Network Architecture (SNA) Communications protocol users can access NOS/VE and NOS applications from an IBM 3270 Information Display System connected to an SNA network. CDCNET supports the 3270 SNA protocol through a terminal interface program. The terminal interface program supporting the protocol communicates with an IBM host passthrough facility called the Host Command Facility (HCF).

Since CDCNET's implementation of the 3270 SNA Communications protocol follows the same conventions as other CDCNET-supported protocols, this appendix supplements the existing CDCNET documentation. The following sections describe only the differences and unique capabilities of the terminal interface program supporting the 3270 SNA protocol. The commands, attributes, and features described in general CDCNET documentation also apply to connections made through a 3270 SNA link, except where noted in this appendix.

Section	Description
The Protocol	Describes general features of the 3270 SNA Communications protocol as supported by the 3270 SNA terminal interface program.
The Environment	Illustrates the 3270 SNA interconnection with CDCNET.
Attribute Support	Describes the subset of CDCNET terminal and connection attributes that the protocol supports.
Program Attention Key Processing	Describes the program attention key processing differences between virtual line mode and transparent mode input editing.

The Protocol

The 3270 SNA Communications protocol, as supported by the 3270 SNA terminal interface program (SNA3270_TIP), provides IBM 3270 Information Display System users access to CDCNET through an SNA network. This section discusses the following areas of difference between the asynchronous protocol and the 3270 SNA protocol, as supported by their respective terminal interface programs.

- Code Translation
- Virtual Line Mode
- Input Processing
- Output Positioning
- Cursor Positioning

Code Translation

The 3270 SNA protocol, as implemented by the 3270 SNA terminal interface program, translates all output from the 7-bit ASCII code set to the EBCDIC code set, and translates all input from the EBCDIC code to the 7-bit ASCII code set. Some EBCDIC characters are not supported by the 3270 terminals. Those output characters that translate to unsupported EBCDIC characters are, instead, translated to null.

Virtual Line Mode

The SNA3270_TIP formats a Virtual Line Mode (VLM) display as a PAGE_LENGTH number of separate lines. The SNA3270_TIP uses 3270 field attribute characters to define the VLM lines on a display. The TIP redefines the entire display whenever the display is cleared or the terminal returns to normal editing.

The first character of each line is reserved for the field attribute character. Therefore, the number of characters allowed on a line is usually one character less than the PAGE_WIDTH setting (that is, 79 out of 80 characters).

Input Processing

During input, the SNA3270_TIP treats each display line separately. The TIP does not combine the entry across display lines into a single message forwarded to the CYBER host. Instead, the TIP processes input entered on multiple lines as multiple input messages. For example, you may type several host commands on separate display lines before pressing the ENTER key. The TIP forwards each command as a separate message to the host. The TIP forwards lines from the top to the bottom of the display.

Output Positioning

The SNA3270_TIP starts VLM output on the next line following the last cursor position at input. Generally, output follows the last input line in a fashion similar to output on an asynchronous terminal. However, you are free to position the cursor anywhere on the VLM display. The SNA3270_TIP positions its output to wherever you desire. For example, you may choose to enter all VLM input on the last line, position the cursor on the home line at the top of the display, and then press ENTER. In this case, the TIP begins all output on the second display line even though the commands were entered at the last line.

Cursor Positioning After Output

The SNA3270_TIP cursor positioning is similar to that provided by the asynchronous TIP. If the last line of output in a complete message repositions the cursor to the next line, the TIP clears the next line and places the cursor on the second column of the next line. If the last line of output does not reposition the cursor, the TIP defines a new input field on the same line as the last output, clears the new field, and positions the cursor at the beginning of the new field. In effect, the last line of output consists of two fields with a blank character before the first field and between the two fields. Neither blank character can be overwritten. Since the last line of output is usually an input prompt, positioning the cursor in the second field allows you to type on the same line as the prompt.

The Environment

Figure J-1 illustrates the 3270 SNA environment. The SNA implementation requires a network DI running 3270 SNA gateway software be connected between the SNA network and CDCNET.

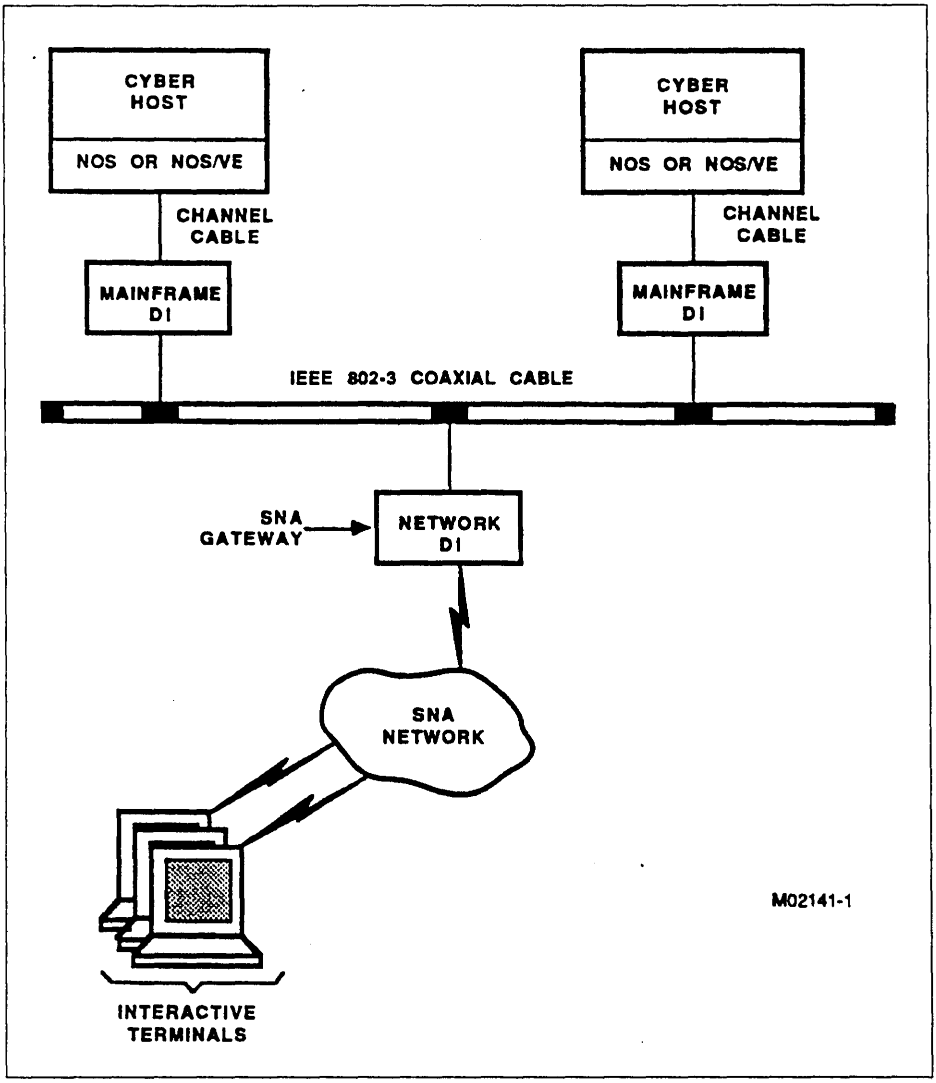


Figure J-1. 3270 SNA Environment

Attribute Support

You can change and display certain terminal and connection attributes from your display station.

Terminal Attributes

The 3270 SNA Communications protocol, as supported by the SNA3270_TIP, supports a subset of the terminal attributes supported by the ASYNCTIP. The default settings for the attributes supported by SNA3270_TIP are displayed in figure J-2.

```
Cancel_Line_Character          :NUL
End_Page_Action                :FFS
Hold_Page                      :ON
Hold_Page_Over                 :ON
Network_Command_Character      :%
Response_Action                :SEND
Status_Action                  :SEND
Terminal_Model                 :IBM_3270
```

Figure J-2. 3270 SNA Communications Protocol Terminal Attribute Defaults

For terminal attributes that the 3270 SNA Communications protocol treats differently from the Asynchronous protocol, see table J-1. The table compares the way the SNA3270_TIP uses these terminal attributes with the way the ASYNCTIP uses the attributes as described in chapter 3.

Table J-1. Terminal Attribute Differences

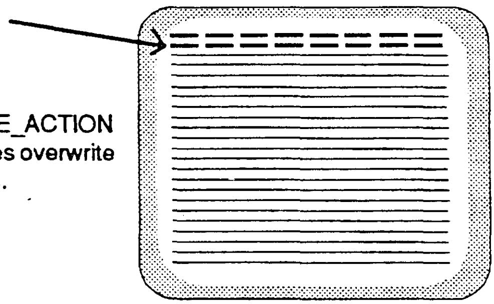
Terminal Attribute (Abbreviation)	Default Setting	Chapter 3 Comparison
CANCEL_LINE_CHARACTER (CLC)	NUL	Any character that can be generated by a 3270 terminal except field mark. Field mark is used as a break key character. See Break Key and Attention Character Actions in this appendix.
END_PAGE_ACTION (EPA)	FFS	NONE or Form Feed Sequence. See Using END_PAGE_ACTION, following this table.
HOLD_PAGE (HP)	ON	Same description applies.
HOLD_PAGE_OVER (HPO)	ON	Same description applies.
NETWORK_COMMAND_CHARACTER (NCC)	%	Any character that can be generated by a 3270 terminal except field mark. Field mark is used as a break key character. See Break Key and Attention Character Actions in this appendix.
RESPONSE_ACTION (RA)	SEND	Same description applies.
STATUS_ACTION (SA)	SEND	Same description applies.
TERMINAL_MODEL (TM)	IBM_3270	Same description applies.

Using END_PAGE_ACTION

For Virtual Line mode (VLM) output, the 3270 SNA Communications protocol supports either an END_PAGE_ACTION (EPA) of NONE or the Form Feed sequence (FFS). When EPA is set to NONE, the SNA3270_TIP overwrites the existing display lines with new output. The overwriting begins at the top of the screen and continues to the bottom of the screen as new lines are input or output. The resulting display consists of lines from the previous display page at the bottom of the screen and lines for the current display at the top.

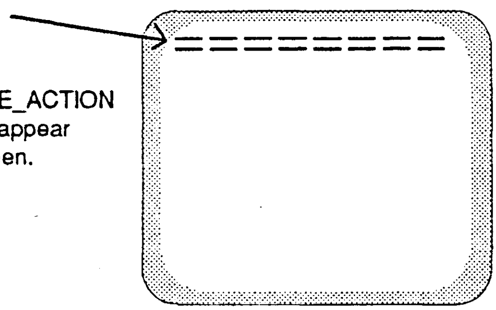
When EPA is set to FFS, the SNA3270_ TIP clears the 3270 screen before writing output to the top of the screen. Under this formatting only the current display page appears on the screen.

New Lines



When END_PAGE_ACTION is NONE, new lines overwrite any previous lines.

New Lines



When END_PAGE_ACTION is FFS, new lines appear on an erased screen.

Page Length and Widths

The 3270 SNA Communications protocol supports page lengths of 12, 24, 32, and 43 lines. The SNA3270_TIP determines the page length for a terminal when the terminal connects to CDCNET. The TIP uses the same page length that the terminal uses to communicate with HCF. You cannot change the page length while connected to CDCNET.

The SNA3270_TIP supports only a fixed page width of 80 columns because HCF supports only 80-column terminals. The TIP does not support the 3270 alternate screen size for terminals that provide alternate sizes. You cannot change page width.

Connection Attributes

The 3270 SNA Communications protocol as implemented by SNA3270_TIP supports a subset of the available connection attributes supported by the ASYNCTIP. The default settings for these attributes are displayed in figure J-3.

```
Break_Key_Action          :0
Input_Block_Size          :160
Input_Editing_Mode        :NORMAL
Input_Output_Mode         :UNSOLICITED
```

Figure J-3. 3270 SNA Communications Protocol Connection Attribute Defaults

The connection attributes are used as described in chapter 4.

INPUT_OUTPUT_MODE (IOM) Differences

The 3270 SNA Communications protocol supports all INPUT_OUTPUT_MODES: FULLDUPLEX (F), SOLICITED (S), and UNSOLICITED (U). The SNA3270_TIP forwards input for each mode to the CYBER host application in the same way as the Asynchronous TIP does. (See INPUT_OUTPUT_MODE in chapter 4). For fullduplex or unsolicited mode, the TIP positions the cursor on the next line after input and reenables input until output is available from the CYBER application. For solicited mode, the TIP neither repositions the cursor nor reenables input until the CYBER application solicits more input by completing its output. Once solicited, all input that has been entered is forwarded to the application.

However, because the 3270 terminal operates differently than asynchronous terminals, in most cases the differences between the various INPUT_OUTPUT_MODES are lost to the 3270 terminal user.

The 3270 operates only in a block mode. That is, the terminal sends the data for an input message all at once rather than character-by-character as keys are pressed. The terminal sends the complete input message when you press the ENTER key, or press either the PF or PA key. If the 3270_TIP is sending output to a terminal, input is inhibited and the terminal locks its keyboard to prevent key entry that would be overwritten by output. Thus, you are unable to interrupt output in unsolicited mode, enter input to be forwarded, or enter network commands while unsolicited ing sent.

Should you begin key entry during a pause in output, however, the SNA3270_TIP halts output for all modes until you complete the input. At that point, all three INPUT_OUTPUT_MODES appear essentially the same.

Break Key and Attention Character Actions

The 3270 SNA Communications protocol recognizes either the Program Access 1 (PA1) key or the field mark character as a break key. To be recognized as a break key, the field mark character must be the first character you enter as input.

When you press the PA1 key or begin input with a field mark character, the SNA3270_TIP performs the user interrupt specified by the BREAK_KEY_ACTION (BKA) connection attribute (see chapter 4).

A 3270 terminal always forwards the PA1 key by itself to the SNA3270_TIP. Therefore, break key sequences (for example, trusted path sequence) require you to perform a two-step sequence when you use the PA1 key. That is, you must first press the PA1 key, wait for the SNA3270_TIP to reenable input (the 3270 input inhibited status to clear), and then enter the key sequence as a second input.

On the other hand, the 3270 terminal forwards the field mark character with any other input you enter with it. In this case, you enter the break key sequence as a single input; the field mark followed by the key sequence.

When invoking a break action, a break key sequence should be the only data forwarded in the input field. For example, no data is allowed to follow a fieldmark `c` in the same field.

NOTE

For some terminals, the IBM host resident HCF program does not forward the PA1 key. Instead, when the PA1 key is pressed, the HCF program forwards the Attention signal (see Attention Signal in this appendix). In this case, you cannot use the PA1 key as a break key.

The 3270 SNA Communications protocol (or the SNA3270_TIP) does not support an attention character.

Special Break Sequences

The SNA3270_TIP examines the characters following a break key for one of the following special sequences:

- Secure access sequence
- Terminate transparent input sequence
- Escape to $NET connection sequence

These special character sequences, which cannot be altered, are described in the following sections.

Secure Access Sequence

The secure access sequence is entered as follows:

1. Press the PA1 key or the field mark character.
2. If you used the PA1 key, wait for input to be reenabled.
3. Press A and then T (lowercase or uppercase), and then clear the remainder of the field.
4. Press the ENTER key.

The SNA3270_TIP terminates all existing data connections.

Terminate Transparent Input Sequence

The terminate transparent input sequence is entered as follows:

1. Press the PA1 key or the field mark character.
2. If you used the PA1 key, wait for input to be reenabled.
3. Press X (uppercase or lowercase), and then clear the remainder of the field.
4. Press the ENTER key.

The SNA3270_TIP terminates transparent input.

Escape to $NET Connection Sequence

The escape to $NET connection sequence is entered as follows:

1. Press the PA1 key or the field mark character.

2. If you used the PA1 key, wait for input to be reenabled.

3. Press AC or C (lowercase or uppercase), and then clear the remainder of the field.

4. Press the ENTER key.

The SAN3270_TIP changes your working connection to the $NET connection.

Attention Signal

The 3270 terminal provides an Attention key that you can use for signaling to the TIP. You can use the Attention key when all other keyboard entry is inhibited. The IBM resident HCF program does not process an Attention signal for terminals connected to CDCNET, but forwards the signal to the SNA3270_TIP for processing. Thus, you can use the Attention key to signal the TIP for special processing.

IBM applications, in general, treat an Attention signal as a request to enable input. The SNA3270_TIP also treats the Attention signal as a request to enable input. If the TIP is outputting data in Virtual Line mode (VLM) when it receives an Attention signal, the TIP responds by holding output at the next end of page. The TIP clears the last display line, positions the cursor at the last line, and enables input as though the HOLD_PAGE attribute were set. In effect, the Attention signal acts as a *one shot* HOLD_PAGE. If VLM output completes before the next end of page, the TIP ignores the Attention signal and reenables input as part of its normal end-of-output processing.

If the SNA3270_TIP is outputting data in Transparent mode when it receives an Attention signal, the TIP suspends output and reenables input at the end of the next transparent output block. If the TIP has no output to send when it receives an Attention signal, the TIP sends an unlock keyboard command to the 3270 terminal. Thus, the unlock keyboard command enables input should a poorly behaving transparent application leave input inhibited at the end of output.

Program Attention Key Processing

The Program Access (PA) and Program Function (PF) keys on the 3270 terminal keyboard are called Program Attention keys. These keys are similar in function to function keys on asynchronous terminals. The SNA3270_TIP processes these keys differently for Virtual Line mode than for Transparent mode input editing. The tables J-2 and J-3 summarize the actions taken by the SNA3270_TIP for Program Attention Keys in Virtual Line mode and Transparent mode, respectively. As noted above, the PA1 functions can also be initiated by entering a field mark character.

Table J-2. Program Attention Key Processing, Virtual Line Mode

Key	Action Taken
PA1	Break_Key_Action (BKA).
PA2	Forwards an *empty* message to the host application.
PA3	Forwards an *empty* message to the host application.
PA4	Forwards an *empty* message to the host application and reformats the 3270 display.
PF	Treats entry as ENTER key.

Table J-3. Program Attention Key Processing, Transparent Mode

Key	Action Taken
PA1	Break_Key_Action (BKA). If BKA is followed by AX, terminates transparent input.
PA2	Forwards an *empty* message to the host application.
PA3	Forwards an *empty* message to the host application.
PA4	Forwards key input to the host application as part of the transparent message.
PF	Forwards key input to the host application as part of the transparent message.

Index

A

B

I

K

L

T

U

V

W

X

Comments (continued from other side)

ɔld on dotted line;
es with tape only.

FOI

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

CDCNET Terminal Interface Usage — 60463850 F

We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other ____________________

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up
- ☐ Other ____________________

What programming languages do you use? ____________________

How do you like this manual? Answer the questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Do you use this manual frequently?

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply: ☐

Name ____________________ Company ____________________

Address ____________________ Date ____________________

Phone ____________________

Please send program listing and output if applicable to your comment.

Quick Reference Index

Quick Reference Index

CONTROL DATA