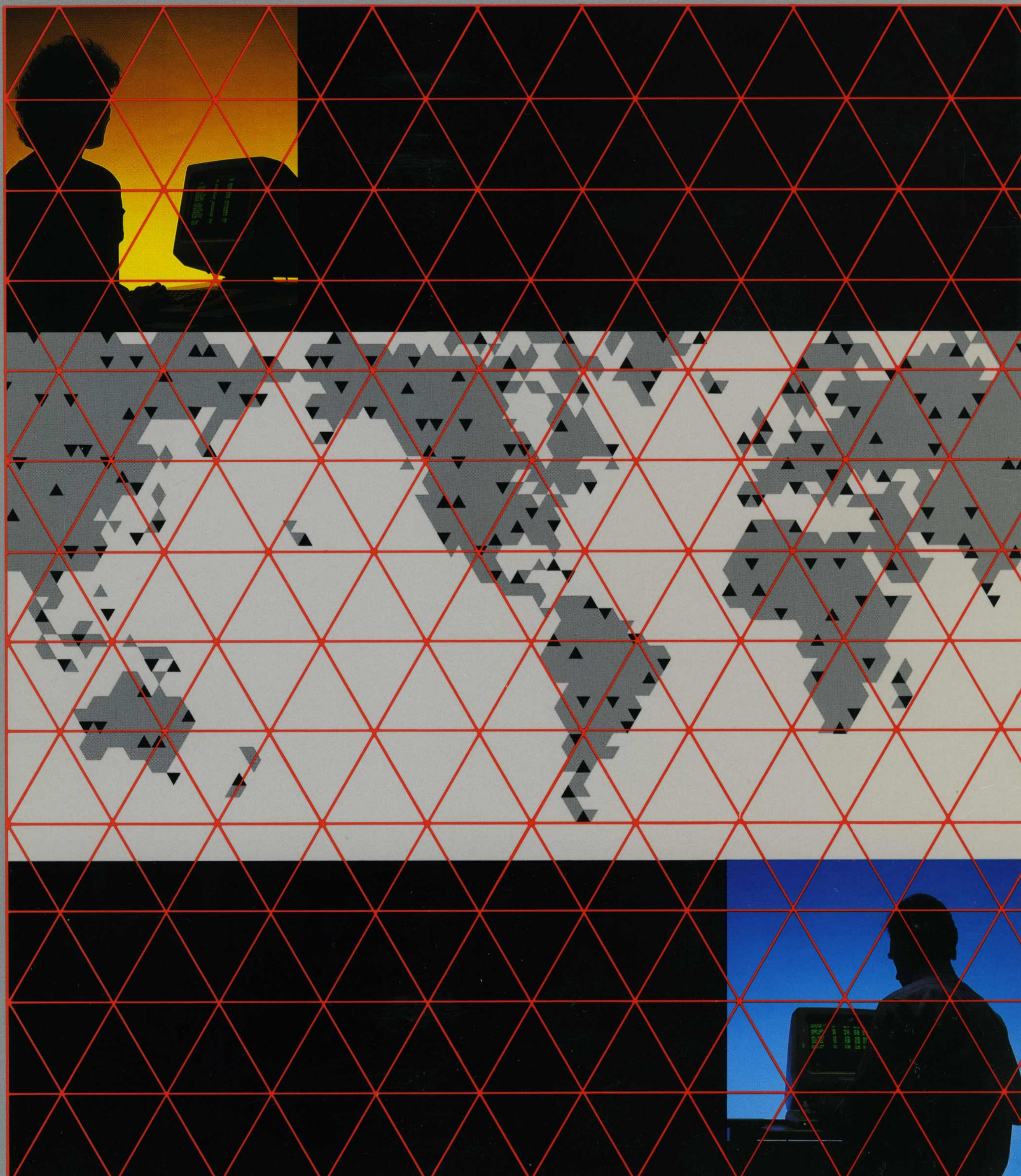
CDCNET

Network Operations Manual



60461520

CONTROL DATA

NOTE

The Network Operations Commands Index is located at the end of this manual. The index contains a list of all CDCNET network operations commands and the page on which each is described.

CDCNET

Network Operations Manual

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 60461520

Manual History

This manual is revision D, printed in September 1987. It reflects CDCNET version 1.2.5 at PSR level 688, for operation on NOS version 2.5.3 and NOS/VE version 1.2.3.

Previous Revision	System Version PSR Level	Date
A	1.0 / 647	December 1985
B	1.1 / 664	September 1986
C	1.2 / 678	April 1987

Contents

Figures

Tables

About This Manual

This manual describes the functions, procedures and commands associated with network operations of a CONTROL DATA® Distributed Communications Network (CDCNET) Version 1.2.5. CDCNET Version 1.2.5 is used with the following operating system software versions: CDC® Network Operating System/Virtual Environment (NOS/VE) Version 1.2.3 and CDC Network Operating System (NOS) Version 2.5.3 or subsequent NOS/VE and NOS versions. CDCNET Version 1.2.5 will run on the following computer systems/models: CDC CYBER 180 Computer Systems, models 810, 830, 835, 840, 845, 850, 855, 860, 930, and 990; CDC CYBER 170 Computer Systems, models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 815, 825, 835, 845, 855, 865, and 875; CDC CYBER 70 Computer Systems, models 71, 72, 73, and 74; and CDC 6000 Computer Systems.

Audience

This manual is written for the person who will perform CDCNET network operations activities, such as starting and stopping communication lines and displaying operational status of network components. It may also be used by communication support analysts, who will use some of the commands described within during troubleshooting. Customer engineers may also use this manual for reference. The reader should have knowledge of NOS/VE and/or NOS concepts and operations, as well as an understanding of CDCNET's general purposes and concepts, as described in the CDCNET Conceptual Overview.

Organization

This manual presents CDCNET network operations concepts and guides you through the first steps of network operations. It describes network operations commands, shows the console and terminal displays these commands generate, and provides suggestions for handling network problems.

This manual is divided into two parts: tutorial and reference. The tutorial part describes the concepts and activities involved in CDCNET network operations. Chapters 1 through 5 are tutorial chapters.

Chapter 1 gives you an overview of CDCNET from a network operator's perspective. You will learn about your role in the network, concepts important to you as a network operator, as well as the kinds of activities you may perform during operations.

Chapter 2 describes the Network Operator Utility (NETOU), which you use to monitor, control, and dynamically reconfigure CDCNET. NETOU is described for both NOS/VE and NOS environments.

Chapter 3 describes the commands and procedures used to control your operations sessions.

Chapter 4 is divided into basic and advanced activities, and presents network operations activities and the command sequences used to perform them.

Chapter 5 is a set of troubleshooting guidelines. It briefly describes your troubleshooting responsibilities, common network problems you may encounter during operations, and actions you can take to contain these problems while keeping the network running. This chapter can be used as a starting point when you are troubleshooting the network. For further investigation of problems, your site should use the CDCNET Network Troubleshooting Guide and, if necessary, the CDCNET Analysis manual.

The reference part contains descriptions of commands used to perform CDCNET network operations. The conventions used in the command descriptions are explained on the divider page for part 2. Chapters 6 and 7 present the session control commands for NOS/VE and NOS environments. Chapter 8 provides detailed descriptions and examples of commands used to control CDCNET. Chapter 9 describes commands used to invoke utilities used for network operations. The commands are in alphabetical order.

Appendixes include a classification of CDCNET commands according to the kinds of activities they perform, and a set of procedures that NOS/VE sites can use to enhance the CDCNET operations environment.

Conventions

The terms "logic board" and "board" are used interchangeably in this manual. They refer to any of the printed circuit board assemblies housed in the device interface (DI), such as the processor board, memory boards and line interface modules.

The terms Ethernet[1] and IEEE 802.3 are used interchangeably in CDCNET manuals. Ethernet refers to a network standard developed by Xerox, Intel, and DEC (Digital Equipment Corporation). IEEE 802.3 is the IEEE adaptation of that standard. The term IEEE 802.3 is a more precise label for the network standard. However, many network operations commands and software programs use the term Ethernet. CDCNET products covered by these standards are compatible with both IEEE 802.3 and Ethernet V.2.

The NOS 2 Operations and Analysis handbooks use the term COP (CDCNET Operator), which is the type of network operator described in this manual.

When descriptions and procedures apply to both a mainframe device interface (MDI) and a mainframe terminal interface (MTI), the term MDI is used for both device interface types. If it is necessary to specify both MDIs and MTIs in a section, they are specified in the initial instance, but from then on, only MDI is used.

CDCNET network operations commands follow the syntax rules for System Command Language (SCL) for NOS/VE, as described in the SCL for NOS/VE Language Definition manual. Abbreviations of commands are accepted. Exceptions to the SCL for NOS/VE syntax in this manual are the NLTERM and NLLIST Utilities used for NOS environments.

Commands and parameters that must be entered as listed are shown in UPPERCASE CHARACTERS. Variable parameters and values are shown in all lowercase characters. Required parameters are shown in **boldface** and optional parameters in *italics*.

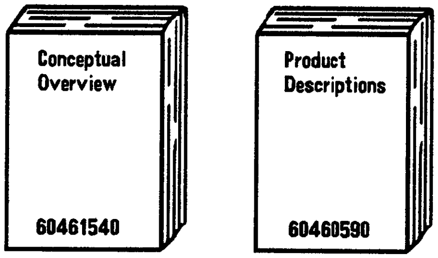
The format of the command descriptions is on the blue divider preceding chapter 6.

All numbers in this manual are decimal (base 10) unless specifically identified as octal (base 8) or hexadecimal (base 16).

1. Ethernet is a registered trademark of the Xerox Corporation.

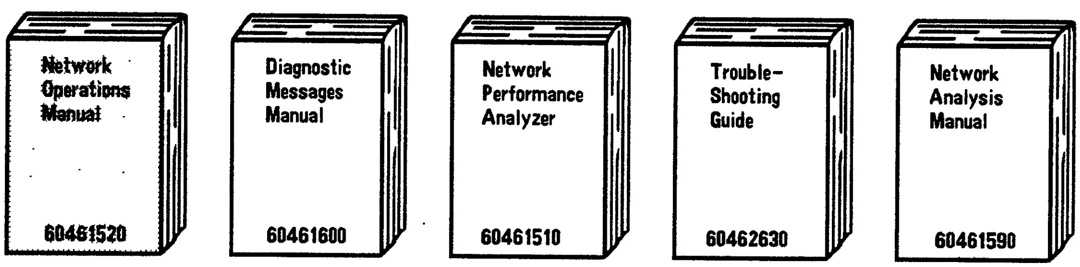
Related Manuals

Background (access as needed):



Installation and checkout manuals:

Operating and troubleshooting manuals:



Additional Related Manuals

The following manuals contain helpful information.

Manual	Publication Number
CDCNET Terminal Interface Usage	60463850
CDCNET Access Guide	60463830
CDCNET Terminal Interface Quick Reference	Online
CDCNET Systems Programmer's Reference Manual, Volume 2 Network Management Entities and Layer Interfaces	60462420
CDCNET Batch Device User Guide	60463863
Remote Batch Facility Reference Manual	60499600
System Command Language (SCL) for NOS/VE Language Definition	60464013
System Command Language (SCL) for NOS/VE System Interface	60464014
NOS Version 2 Reference Set, Volume 3	60459680
NOS Version 2 Reference Set, Volume 4	60459690
NOS Version 2 Operations Handbook	60459310
NOS Version 2 Analysis Handbook	60459300
NOS/VE System Analyst Reference Set Network Management	60463916

Ordering Manuals

Control Data manuals are available through Control Data sales offices or Control Data Literature and Distribution Services, 308 North Dale Street, Saint Paul, Minnesota, 55103.

Submitting Comments

Control Data welcomes your comments about this manual. Your comments may include your opinion of the usefulness of this manual, your suggestions for specific improvements, and the reporting of any errors you have found.

You can submit your comments on the comment sheet on the last page of this manual. If the comment sheet has already been used, mail your comments on a separate sheet of paper to:

Control Data Corporation
Technology and Publications Division
ARH219
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

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 `Report a new problem or change in existing PSR` from the main SOLVER menu.

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

3. Select `Write a comment about a manual` from the new menu.

4. Respond to the prompts.

Please indicate whether you would like a written response.

Acronyms

A-to-A	Application-to-Application
ANACD	Analyze CDCNET Dump
ARP	Address Resolution Protocol
CCL	CYBER Command Language
CDCNET	Control Data Distributed Communications Network
CIM	Communications Interface Module
DCNS	Distributed Communications Network Software
DI	Device Interface
DDN	Defense Data Network
DOD	Department of Defense
EGP	Exterior Gateway Protocol
ESCI	Ethernet Serial Channel Interface
FTP	File Transfer Protocol
IAF	Interactive Facility
I/O	Input/Output
IP	Internet Protocol
LIM	Line Interface Module
MANCC	MANAGE_CDCNET_CONFIGURATION
MCI	Mainframe Channel Interface
MDI	Mainframe Device Interface
ME	Management Entity
MPB	Main Processor Board
MTI	Mainframe Terminal Interface
NAM	Network Access Method
NAM/VE	Network Access Method/Virtual Environment
NDI	Network Device Interface
NETFS	Network File Server
NETOU	Network Operator Utility
NOS	Network Operating System

NOS/VE	Network Operating System/Virtual Environment
NP	Network Products
NP IVT	Network Products Interactive Virtual Terminal
NPA	Network Performance Analyzer
NTF	Network Transfer Facility
PMM	Private Memory Module
REFCLF	REFORMAT_CDCNET_LOG_FILE
SCL	System Command Language
SMM	System Main Memory
T-to-A	Terminal-to-Application
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDI	Terminal Device Interface
TDP	Terminal Definition Procedure
TIP	Terminal Interface Program
TUP	Terminal User Procedure
URI	Unit Record Interface

Part 1: Tutorial

Introduction to Network Operations 1

Introduction to Network Operations 1

This chapter gives an overview of the CDCNET network operations process and explains CDCNET concepts you should know when performing network operations.

It is important that you read this chapter before logging into CDCNET for network operations, or before entering any commands described in this manual. For more information about CDCNET software and hardware concepts and terminology, review the CDCNET Conceptual Overview manual and the CDCNET Product Descriptions manual.

CDCNET is a distributed data communications network and a collection of data communications equipment interconnected by communications channels. CDCNET distributes its automated communications control and network management functions throughout the network, using a collection of device interfaces (DIs). DIs are connected to mainframes, asynchronous terminals and printers, batch input and output equipment, and other networks. DIs may be connected to communications media that carry information formatted for CDCNET from one DI to another.

CDCNET may have a variety of configurations, depending upon the size of the network, number of terminals the network supports, and the amount of communications traffic the network generates.

Network Operations Concepts

The following are concepts you should read and understand before performing network operations.

Host Computer

A host computer consists of a mainframe computer and its operating system. Together, they provide applications and services to the computer network. For CDCNET to operate, it must have at least one CDC mainframe running Network Operating System/Virtual Environment (NOS/VE) or Network Operating System (NOS); one mainframe device interface (MDI) or one integrated communications adapter (ICA); and one terminal device interface (TDI) or mainframe terminal interface (MTI). The CDC mainframe acts as the network host. As a host, the mainframe can download software to DIs, provide programs to configure the network, and run other utilities needed by CDCNET, such as the utility that analyzes the CDCNET log file.

Device Interface

A DI is the main hardware device used to implement CDCNET. The DI controls access to the network and controls data communications through the network. Both DI hardware and software are modular. The type of hardware and software housed in a DI depends on the DI's specific function as a network communications controller. For more information about DI hardware and software, refer to the CDCNET Conceptual Overview and the CDCNET Product Descriptions manuals.

Lines, Trunks, and Network Solutions

You control several types of network communications media in CDCNET network operations; they are described in the following section.

Communication Line

A communication line connects data terminating equipment (DTE), such as a terminal or printer, to a DI. The data carried on this line from a DI is meant specifically for the terminal device, or is sent from the terminal device to the DI to which it is connected. Unlike a network solution, a line does not receive data meant for other areas of the network.

The DI hardware that controls communication lines includes the Communications Interface Module (CIM), Line Interface Modules (LIMs), and Unit Record Interface (URI) LIMs. The DI software that controls communication lines and the input to and output from terminal devices is called a terminal interface program (TIP).

A communication line is defined by the DEFINE_LINE command in a DI's system configuration procedure.

Trunks

A trunk carries data for many devices connected to the network that may or may not be attached to the trunk. A trunk may be the underlying medium for a network solution. A trunk may also be the medium used to connect to a Public Data Network (PDN) through a gateway that acts as a translator between different protocols. The physical device for a trunk may be an Ethernet coaxial cable, a NOS/VE or NOS host's mainframe channel, a high-level data link control (HDLC) line, or an X.25 communication line.

A trunk is defined by a DEFINE_xxxx_TRUNK command, where the xxxx portion may be ETHER, CHANNEL, HDLC, or X25.

Network Solution

Network solutions interconnect two or more CDCNET DIs, using CDNA protocols. A network solution is a trunk that has been configured to carry both user data and CDCNET network management traffic.

A network solution is the main structural element of a CDCNET-type network. It can carry data from any point in the CDCNET network to any other point in the network. Unlike trunks and lines, it can also carry CDCNET network management services (such as log messages and alarms) and other services provided by the network (such as connections to host services).

A network solution is defined by a DEFINE_xxxx_NET command, where the xxxx portion may be ETHER, CHANNEL, HDLC, or X25.

Catenet

A catenet is a group of connected network solutions. This term is often used in commands and text when referring to all the DIs and network solutions in a site's network.

Logging Group

A logging group is a subset of DIs, within a catenet, that send their messages to a common log file. A logging group is established at configuration time. Each DI can belong to only one logging group. At configuration time, you can assign each DI the name of the logging group to which the DI belongs. The default logging group name on the configuration commands is CATENET. You can also configure each DI with the list of message numbers identifying the log messages it can send to the log file. Enable the default set of log messages by entering the DEFINE_SOURCE_LOG_GROUP command without the message parameter.

The Network Operator Utility

The Network Operator Utility (NETOU) supports the set of commands and features used to monitor, control, and logically reconfigure CDCNET. Chapter 2 describes NETOU.

Operations Station

This manual uses the term "operations station" to refer to the remote terminal or host console from which operations activities are performed through NETOU.

Commands

NETOU supports commands to control the network from your operations station. Operations commands can be divided into the following types:

- Session control commands. These are commands that define and control your operations environment and operations sessions, but do not control or change the network. Since there are different operations environments for CDCNET on NOS/VE and NOS, each operating system has its own set of session control commands and activities. Directions for using these commands are in chapter 3. Command descriptions are in chapter 6 (NOS/VE session control commands), and chapter 7 (NOS session control commands).

- Network commands. These commands are used to monitor, control, and dynamically change the logical definition of network equipment (see Network Configuration, this chapter). Directions for using many of these commands are in chapter 4. Command descriptions are in chapter 8.

Network Operations Activities

As a network operator, you control the network by managing the network's DIs and other network components, such as network solutions, communication lines, gateways, and by monitoring and responding to alarms and other messages generated by the network. These activities are performed by sending commands to DIs and observing the command responses.

You monitor, control, and occasionally change the logical configuration of CDCNET. You can perform these activities either from an interactive terminal or from a host computer console. Network operations commands are equivalent whether you perform network operations from an interactive terminal or host console. Chapters 2, 3, and 4 are tutorial chapters that show you how to run network operations sessions on NOS/VE or NOS.

The network activities you perform may vary depending on your site's configuration and communication needs. You may perform some activities more often than others, again depending on your site. In this manual, four types of network operations activities are described: session control, basic network operations, advanced network operations, and troubleshooting.

Session Control Activities

Session control involves setting up and controlling your operations session. Examples of session control include controlling which DIs send alarm messages to your operations station, and routing NETOU command responses to a file that will serve as a record of the responses. These activities do not actually control or change the network.

Basic Operations Activities

Basic operations activities are operations you are likely to perform on a regular basis. These activities do not require special training beyond the scope of this manual, such as training in software analysis or a complete understanding of the network's logical configuration. Nevertheless, basic activities perform an important role in running and maintaining the network. Some basic activities include starting and stopping communications on communications lines, sending messages to terminal users, and synchronizing DI time clocks. Basic operations activities are described in the first part of chapter 4.

Advanced Operations Activities

Advanced operations activities require a more thorough understanding of the network, its configuration, and the CDCNET software that runs in DIs and on the host computer. Some advanced activities use several different programs that are not a part of NETOU. Other advanced activities can have a major effect on the network's performance, such as shutting off a DI or changing the network's logical configuration. Because advanced operations activities can affect the network's performance, your site may choose to have an analyst perform them, or to have you perform the activities under an analyst's supervision. Advanced operations activities include starting and stopping a gateway, stopping a DI, making online network configuration changes, and loading and unloading CDCNET software. Advanced activities are described in the second part of chapter 4.

Troubleshooting

When problems occur in the network (such as users being unexpectedly disconnected from host services), CDCNET network commands can be used as a first step in troubleshooting. Network commands can be used to gather information about a problem and to isolate failures. Chapter 5 contains guidelines you should follow when troubleshooting. Depending on the situation, you may be able to fix the problem yourself, using the available operations commands, or you may have to refer the problem to an analyst or customer engineer (CE). The guidelines in this chapter should help to determine the seriousness and extent of the problem. Consult the CDCNET Troubleshooting Guide for more information on troubleshooting procedures.

Gateways

A gateway is a program which connects two networks that use different protocols. A gateway acts as a protocol converter to enable systems with unlike protocols to communicate with each other. CDCNET's Control Data Network Architecture (CDNA) is a group of protocols. For a network using CDNA protocols to communicate with another network that uses different protocols, the two networks must communicate through a gateway program. Some gateways used in CDCNET are used only for NOS hosts; these are the Network Products gateways.

Network Products Gateways (NOS Only)

NOS supports a network based on 2550 Network Processing Units. This network has been known by various names, including Network Products and Network Host Products. The Network Products gateways allow information to be transferred between CDCNET and a non-CDNA NOS host. The Network Products protocol is different from the CDNA protocol; a gateway is necessary for CDCNET to access the NOS host.

Each NOS host uses an MDI or MTI to interface to CDCNET. An MDI or MTI provides the Network Products gateway function. Network Products connections exist between the gateway function in each MDI or MTI and its associated host. MDIs and MTIs containing Network Products gateways are a member of both networks and understand both CDCNET and Network Products protocols. To CDCNET itself, a gateway is seen as the end of the connection, even though a host mainframe is beyond the gateway.

There are two kinds of Network Products gateways: The terminal-to-application (T-to-A) gateway and application-to-application (A-to-A) gateway. The T-to-A gateway is called the Network Products terminal gateway (abbreviated as NP_TERMINAL_GW in network commands). The NP terminal gateway allows both interactive and remote batch terminal users to connect to the NOS host through CDCNET. There are two parts to the NP terminal gateway: The Interactive Virtual Terminal gateway (IVT gateway) and the Remote Batch Facility gateway (RBF gateway). The batch gateway is dependent on the interactive gateway. The NP terminal gateway software resides in an MDI or MTI. This gateway is an important portion of DI software. If the gateway is logically deleted or if the gateway software is removed from a DI, terminal users cannot connect to a NOS system.

The NP A-to-A gateway (abbreviated as NP_GW in network commands) is a gateway that allows applications on another NOS/VE, NOS, or foreign system to access the NOS system. The NP A-to-A gateway also allows applications on the NOS system to access applications on other NOS/VE, NOS, or foreign systems. File transfer (PTF) and job transfer (QTF) are the primary users of the NP A-to-A gateway.

X.25 Gateway

X.25 circuits allow CDCNET to access public data networks, such as Telnet and Tymnet. An X.25 gateway is used to transfer data from a host connected to CDCNET to a host in another network at the other end of the X.25 circuit. The X.25 gateway allows A-to-A connections to take place over an X.25 circuit. Some network commands control an X.25 gateway, and can be used to start and stop access to X.25 services.

TCP/IP Gateway

The TCP/IP gateway supports CDCNET access to Department of Defense (DOD) networks and provides A-to-A services such as FTP. The gateway supports CDCNET access to Defense Data Networks (DDN) or workstations using TCP/IP protocols that support the Advanced Research Project's Agency Network (ARPANET) community. The gateway also supports the Excelan PC. There are network commands to control a TCP/IP gateway and to stop and start TCP/IP services.

Sending Network Commands

The following section describes concepts used to send commands to the appropriate destination in CDCNET.

Network commands must be sent to the network's DIs, affecting DIs and their hardware and software components. For example, there are network commands which display the operational status of a DI's logic boards, control statistics collection, add or delete lines from a network's configuration, stop communications on a network component, or run diagnostics on DI boards and ports.

To send a CDCNET network operations command to a DI, insert the command within another command which acts in the manner of an addressed envelope. This command is called SEND_COMMAND. It sends the network command to a specific DI or list of DIs. Session control commands are not sent to DIs to control network equipment, therefore they do not have to be sent within a SEND_COMMAND.

Command Responses and Alarms

In CDCNET, once a network command arrives at the proper destination, it is processed, and a response to the command is sent back to you.

Some messages are sent to you unsolicited, that is, without sending a command. These unsolicited messages are called alarms. Alarms are messages generated by network software for various events worthy of operator notification which the software detects, or for actions the software takes.

Physical and Logical Names

When sending network operations commands to DIs, you can address DI components (boards, lines, trunks, network solutions, terminal devices) by name. The following naming conventions are allowed.

- Its physical name
- Its logical name

Physical Names

Physical names are given to a DI's hardware devices, such as boards, ports, memory banks, terminal devices, communication lines, network solutions, and the DI itself. With the exception of boards, physical names are used as the default logical names for many DI components with logical names. Logical names are defined by CDCNET configuration commands. Once defined, the logical names are used in place of, and not in addition to, physical names. Some network operations commands, such as the online diagnostics commands, require that you specify physical names of devices.

Physical names begin with a $ character.

The physical name for a DI system is in the form

$DI_system_id

where system_id represents the unique 12-character system ID assigned to the DI. An example of a DI physical name is $DI_0800253000A1.

For DI boards, the physical name is in the form

$devicen

The device portion of the name refers to board type, which may be one of the following values.

MPB	Main processor board.
SMM	System main memory.
PMM	Private memory module.
CIM	Communications interface module.
ESCI	Ethernet serial channel interface.
MCI	Mainframe channel interface.
LIM	Line interface module.
URI	Unit record interface.

SMM bank number (specified as BANK).

LIM port number (specified as PORT).

URI port number (specified as PORT).

The n portion of the name is a number that may have one of the following values.

- Board slot number (0 through 7). Refers to the board slot number of the hardware device in the DI. A DI contains two sizes of boards, large boards (MPB, PMM, SMM, CIM), and small boards (LIM/URI).
- System Main Memory (SMM) bank number (0 through 1).
- LIM port number (depending on the LIM model, either 0 through 1, 0 through 3, or 0 through 7 from top to bottom). Port 0 is the top port on the LIM.
- URI port number (0 through 1). Note that only URI port 0 is currently supported.

The following are examples of physical names for DI boards.

$CIM3	Physical name for CIM board in board slot 3
$ESCI4	Physical name for ESCI board in board slot 4

When a component is a subassembly of a device, such as a port on a LIM, the physical name of the subassembly is a concatenation of the main device name and the subassembly's name, joined by an underscore. For example, $LIM5_PORT2 is the physical name for the second port on a LIM board in LIM board slot 5, $SMM2_BANK0 is the physical name for bank 0 on a SMM board in board slot 2, and $URI7_PORT0 is the physical name for port 0 of a URI board in slot 7.

Figure 1-1 shows how physical names are assigned in a DI and shows an example of how boards may be installed in a DI.

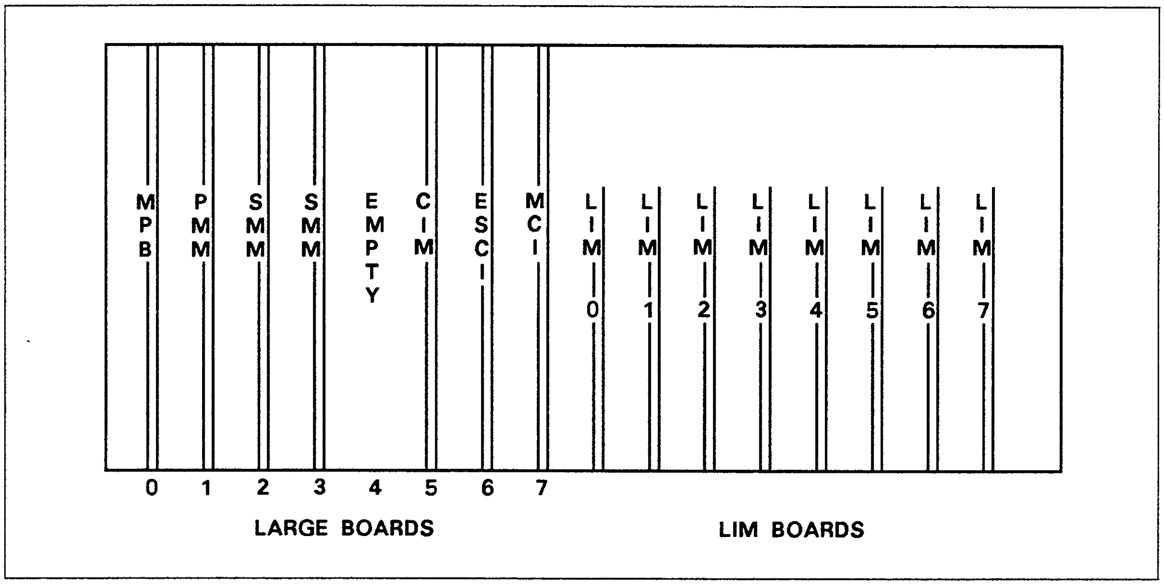


Figure 1-1. Example of Physical Names

Based on the configuration of boards shown in figure 1-1, the following physical names are assigned.

Large Board Physical Names	LIM Physical Names	Port Physical Names
$MPB0	$LIM0	$LIM0_PORT0 to $LIM0_PORT3
$PMM1	$LIM1	$LIM1_PORT0 to $LIM1_PORT3
$SMM2	$LIM2	$LIM2_PORT0 to $LIM2_PORT3
$SMM3	$LIM3	$LIM3_PORT0 to $LIM3_PORT3
$CIM5	$LIM4	$LIM4_PORT0 to $LIM4_PORT3
$ESCI6	$LIM5	$LIM5_PORT0 to $LIM5_PORT3
$MCI7	$LIM6	$LIM6_PORT0 to $LIM6_PORT3
	$LIM7	$LIM7_PORT0 to $LIM7_PORT3

Large board slot 4 is empty, therefore no physical name is assigned for slot 4.

A port's physical name is used as the default logical name for a communication line. For example, the default logical name for a line connected to LIM 0, port 0 on a DI is $LIM0_PORT0.

The physical name for a terminal device is made up of:

- $
- The type of terminal device.
- The last six digits of the twelve-digit hexadecimal system ID to which the terminal is connected.
- The LIM number to which the terminal is connected.
- The port number which is connected to the communication line leading to the terminal device.
- The cluster address for the terminal device.
- The device address for the terminal device.

For example, given a terminal device configuration with the values:

System ID:	08002510003C
LIM number:	4
Port number:	2
Device type:	console
Cluster address:	00
Device address:	01

the terminal device would have the following physical name, which is the default logical name,

$CONSOLE_10003C_420001

Logical Names

Logical names allow you to give descriptive names other than physical names to network components, names which may be more immediately meaningful to your site than the physical names. For example, your site may choose to develop a descriptive naming scheme for communication lines. Defining short, descriptive logical names for DIs will make it easier for you to specify the system name when sending commands to DIs, rather than specifying the entire physical name. If logical names are not defined for components on configuration commands, the default logical name will be the component's physical name. For example, if you do not define a logical name for a line, the line will assume a default name which is the physical name of the LIM and port to which the line is connected. If the line is connected to port 3 on LIM 1, then the default logical name for the line will be the port's physical name: $LIM1_PORT3.

The CDCNET Configuration and Site Administration Guide contains conventions for creating logical names, and a table that shows the construction of logical names. Refer to that manual for more information. The default logical names for network components are shown in the DEFINE command descriptions in the Network Commands chapter in part 2 of this manual.

A NETOU command, DISPLAY_LOGICAL_NAMES, displays the logical names defined for a DI, such as logical names for trunks, network solutions, and communication lines. See the command description in chapter 8.

Example logical names:

Device interface names

North_Bldg_TDI_1

MDI_3C (for a DI with a system ID of 0800251003C)

TDI_134 (for a DI with a serial number of 134)

Trunk names

ESCI3

MCI2

Network names

Network_1

ESCI_Network

Line names

Engineering_Port_1

Line 12 (for a line on $Lim1, Port 2)

Compsci_02

Addresses and Titles

Each DI has a unique address and title that identifies its location in the network. DI addresses are assigned during hardware installation. DI titles are assigned during software configuration. A configuration command (DEFINE_SYSTEM) may be used to define a logical name for the DI. The logical name maps to the DI's title and address. The system title is created from this logical name or from the default logical name if a logical name is not specified. The difference between a title and other logical names known by a DI system is that titles are registered with a service called Directory Management Entity (ME) and may be known throughout the catenet; other logical names such as line names are local to the individual DI system. System titles are known throughout the catenet.

For example, suppose a DI is installed with the system ID of 0800250A1FF2 hexadecimal. This system ID is its system address. During software configuration, the DI is defined as a TDI with the logical name First_Floor_TDI. The system title is then $SYSTEM_FIRST_FLOOR_TDI. You, as the system operator at configuration time, do not actually enter the portion of the system title represented by $SYSTEM. The portion of the system title represented by $SYSTEM is the common prefix for all system titles assigned by convention. The NETOU generates this common prefix automatically. When operations commands are sent to this TDI, the logical name can be specified as the destination of the command and will be interpreted as corresponding to the DI's system address and title, with the command received at the correct destination.

Network operations commands can also be sent to DIs by specifying their default logical names, which are described in the Physical Names section of this chapter.

It is important to keep track of titles, addresses and logical names. For suggestions on maintaining complete and accurate records of titles, addresses, and other network information, see the Recordkeeping section in chapter 4.

Network Configuration

The material in this section is intended to provide background information on the logical and physical configuration processes that ready CDCNET for operations. You do not have to understand the logical configuration process completely in order to perform the tasks described in this manual. Both the logical and physical configuration process should be completed by other site personnel by the time you begin CDCNET network operations. Defining and maintaining your site's initial logical configuration is the responsibility of the site administrator (the site administrator's responsibilities are documented in the CDCNET Configuration and Site Administration Guide. If you need more details on CDCNET logical configuration, refer to that manual).

CDCNET configuration involves planning and installing the network's hardware (physical configuration) and preparing the software used to run the network (logical configuration). Both physical and logical configuration must be completed before CDCNET can be operational.

Physical Configuration

The Local Area Network Installation and CDCNET DI Installation and Checkout manuals explain how device interfaces and other network hardware components, including LAN cables and components (transceivers, repeaters, and multiplexers), are installed. This phase of configuration involves planning the physical layout, installing cables and lines, installing boards in the DIs, connecting the DIs to the network communications media, and ensuring that all the required hardware is present.

Logical Configuration

Logical configuration involves planning and preparing the software which runs in the DIs. The logical configuration is a description of functions of the DI and components connected to it. This description is in the form of configuration commands that define characteristics for the software which runs in the DIs. For example, configuration commands can be used to define the logical names of DIs and trunks and network solutions, to declare the line speeds for communication lines, and to define characteristics of batch devices such as printers and card readers. Configuration commands can also be used to define logical names for network components, such as DIs, trunks, network solutions, communication lines and terminal devices.

Logical configuration is necessary because DIs cannot function if they do not contain the software necessary to perform network tasks and operations.

The CDCNET Configuration and Site Administration Guide describes logical configuration. Logical configuration is the responsibility of a CDCNET site administrator, and should be accomplished prior to your beginning network operations. Occasionally during network operations, you may be directed to change the logical configuration while the network is running. For more information, refer to Changing Network Logical Configurations in chapter 4.

Network Operator Utility (NETOU) Overview 2

Network Operator Utility (NETOU) Overview 2

This chapter describes the Network Operator Utility (NETOU), as used in NOS/VE and NOS environments. Since NETOU has some different features on NOS/VE and NOS, the chapter is divided into three sections: Operations in a NOS/VE Environment; Operations in a NOS Environment; and Common Network Operations Features. The chapter describes how to access and use NETOU, screen displays during network operations, command syntax, how to send network operations commands, and command responses and alarms.

The Network Operator Utility (NETOU)

You perform network operations using the NETOU. NETOU allows you to access CDCNET and perform network operations activities from a remote terminal or from host console on NOS.

The command syntax is the same whether you are at an interactive terminal or host console. However, some aspects of terminal and console command entry, display and screen control are different. This chapter explains these differences.

Operations in a NOS/VE Environment

Figure 2-1, NETOU Operating Environment for NOS/VE, shows the major software and hardware components that provide the operations environment on NOS/VE. For NOS/VE environments, NETOU consists of the CYBER-resident NETOU application, and the Dependent Command Management Entity resident in each DI. On NOS/VE, you log into NOS/VE Timesharing, using the site-defined name for the Timesharing service, and enter a command to invoke NETOU. Selecting NETOU allows you to add the subset of NETOU session and network control commands to the NOS/VE commands you are currently allowed to enter. You may continue to enter other NOS/VE commands during any active session with NETOU.

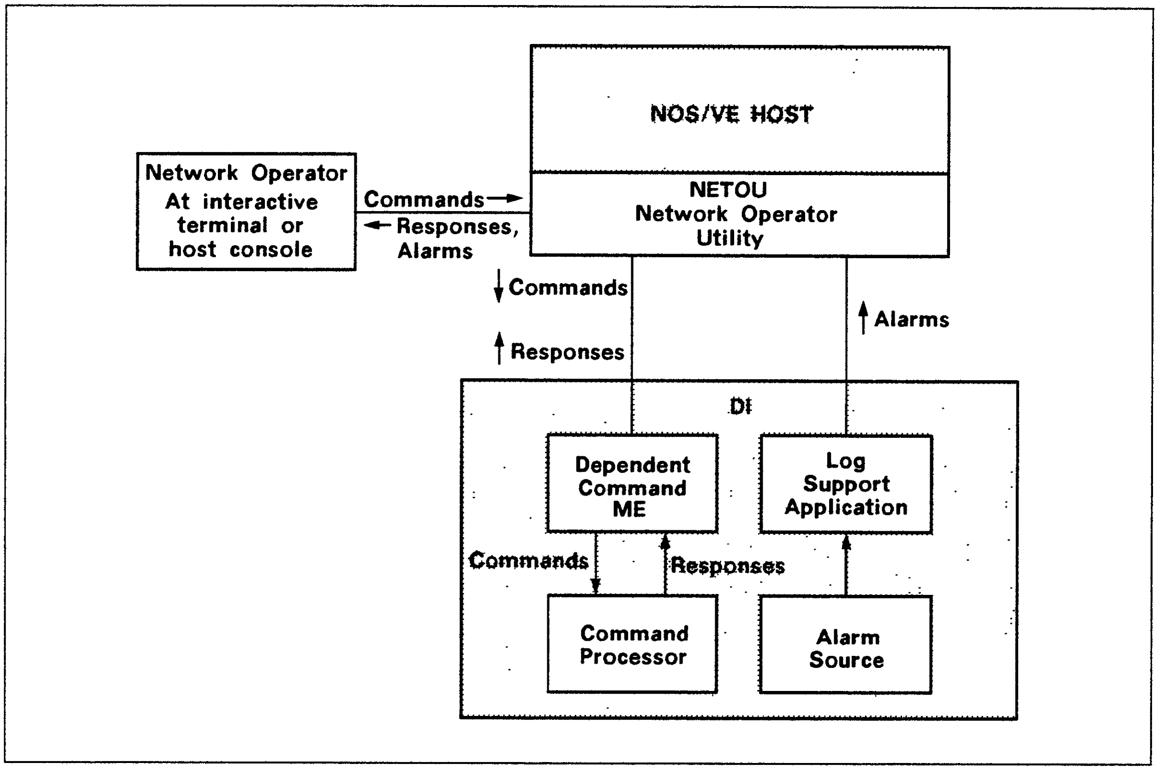


Figure 2-1. NETOU Operating Environment for NOS/VE

Accessing NETOU

Before accessing NETOU, you must connect to a service on the host system to begin an interactive terminal session. Use the CREATE_CONNECTION (CREC) terminal user command, and select the NOS/VE Timesharing service, the title for which is defined at your site through the Manage_Network_Applications Host Utility. Check with the NOS/VE site administrator for the title of the Timesharing service. The following is an example of a connection to the Timesharing service on NOS/VE entitled NVE.

```
create_connection nve
```

If you need to review how to use the CREATE_CONNECTION command, refer to the CDCNET Terminal Interface Usage manual.

To access NETOU, first log in to NOS/VE using the standard NOS/VE login process.

You must be validated to use NETOU. Access to NETOU is controlled by the NOS/VE operating system. Your site's family administrator, through the ADMINISTER USER Utility, controls the NETOU privileges available to you. Check with your site's operating system administrator if you are not sure you are validated to use NETOU. Refer to CDCNET Configuration and Site Administration Manual for more information about NETOU validation.

To use NETOU, enter the following NOS/VE command after you have logged in.

NETWORK_OPERATOR_UTILITY (NETOU or NOU)
PROLOG = file reference
STATUS = status variable

Both parameters are optional. The PROLOG parameter specifies a file containing commands to be executed once NETOU is invoked. Any NOS/VE or NETOU commands can be in the prolog. The default file reference for the PROLOG parameter is $USER.NETWORK_OPERATOR_PROLOG. If you do not specify this parameter and the default prolog file does not exist, no prolog file processing will occur. For more information on prologs, refer to Using a Prolog in the Session Control on NOS/VE section of the Session Control chapter. For information on the STATUS parameter, refer to the basic status concept for NOS/VE SCL in the SCL Language Definition manual.

You may add the NETOU command to your NOS/VE prolog (see the System Access chapter of the SCL for NOS/VE System Interface manual), so that NETOU will be automatically invoked each time you log in.

Prompts for NETOU

The prompt for NETOU is:

```
nou/
```

This prompt indicates that you have selected NETOU and can begin entering NETOU commands. You may also enter other NOS/VE commands.

You may also receive the following prompt for NETOU.

```
nou../
```

This prompt indicates that the previous line you input was continued to another line.

Paging

Paging allows you to move forward within a display on the terminal screen. You may enable or disable paging using the CDCNET terminal command CHANGE_TERMINAL_ATTRIBUTES (CHATA). To do this, enter the network command character (NCC) (shown here as a percent sign [%], but the actual NCC may differ for your terminal), and the CHANGE_TERMINAL_ATTRIBUTES command, as in the example that follows. You may also enter the CHANGE_TERMINAL_ATTRIBUTES command without a preceding network command character to cause the host version of the command to execute.

```
%CHANGE_TERMINAL_ATTRIBUTES HOLD_PAGE=ON or OFF
```

ON enables paging; OFF disables paging. The default is for paging to be OFF. When paging is on, to scroll to the next page of text, enter a carriage return or a control character. For more information about the CHANGE_TERMINAL_ATTRIBUTES command and the network command character, refer to the CDCNET Terminal Interface manual.

NETOU Terminal Display Format

NETOU at a terminal uses virtual line mode format (as opposed to full screen mode) for display output. Commands are entered on a line-by-line basis. Responses are returned in a line-by-line format as well. You will use some utilities to perform network operations tasks that use full screen mode. These utilities run outside of NETOU, and include the Network Performance Analyzer (NPA), and the Manage CDCNET Configuration Utility (MANCC).

Exiting NETOU

To exit NETOU, enter the QUIT command.

```
quit
```

When you enter QUIT, you can exit NETOU and still remain logged in to Timesharing. QUIT removes the NETOU commands from the set of commands you are allowed to enter. The LOGOUT command both terminates NETOU and logs you out of Timesharing.

Entering Network Commands

NETOU commands are valid only within a NETOU session. The session begins when you enter the NETOU command to invoke NETOU. The session ends when you enter the QUIT command. You use the SEND_COMMAND command to send network commands to the appropriate destination.

SEND_COMMAND (NOS/VE Version)

A network command is embedded within SEND_COMMAND as a string value, and another parameter sets the destination for the network command. SEND_COMMAND has the following format on NOS/VE.

SEND_COMMAND
COMMAND = string
SYSTEM = list of name
OUTPUT = file name
STATUS = status_variable

There are two required parameters: COMMAND and SYSTEM. COMMAND is the CDCNET operations command to be sent to the specified DI. The command is entered as a string value enclosed by apostrophes (').

NOTE

If the command you are sending contains any apostrophes, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized. Otherwise, NETOU will assume the embedded apostrophe signals the end of the NETOU command, and errors could result.

For example, the following command contains an embedded apostrophe in the message being transmitted to all terminals connected to TDI1.

```
send_command c='write_terminal_message,..
m=''ENGINEERING''''s network down until 10:00''',..
s=tdi1
```

SYSTEM is the logical or physical DI name or list of DI names to which the command is to be sent. If a CDCNET command is sent to more than one CDCNET system, a response must be received from each system for the command to complete.

The other parameters are optional. Refer to the SEND_COMMAND (NOS/VE version) description in chapter 6 for more information on these parameters.

SEND_COMMAND Example

The following command sequence would be entered to stop traffic on a communications line connected to a DI named TDI_3.

```
send_command command='stop_line line_name=line3',system=tdi_3
```

The actual command to stop communications traffic is enclosed within a SEND_COMMAND command that specifies the DI (TDI_3) to which the line is connected.

Operations in a NOS Environment

Figure 2-1 shows the major software and hardware components that provide the operations environment on NOS. On NOS, NETOU is an application that you select as you would other NOS applications, such as Interactive Facility (IAF). For the NOS environment, NETOU consists of the CYBER-resident NETOU application; the Operator Support Application (also known as the Independent Command ME) which resides in MDIs/MTIs that have been chosen, during logical configuration, to provide operator support; and the Dependent Command ME which is resident in each DI in the network. When you select NETOU, your job is dedicated to NETOU until you exit that application. Commands other than NETOU session and network control commands will not be accepted.

On NOS, NETOU can be used either at a remote terminal or a NOS host console. On the NOS host console, NETOU runs through the NAM K display. The K display has special character and command entry restrictions that are described in the section entitled Network Operations from a NOS Host Console.

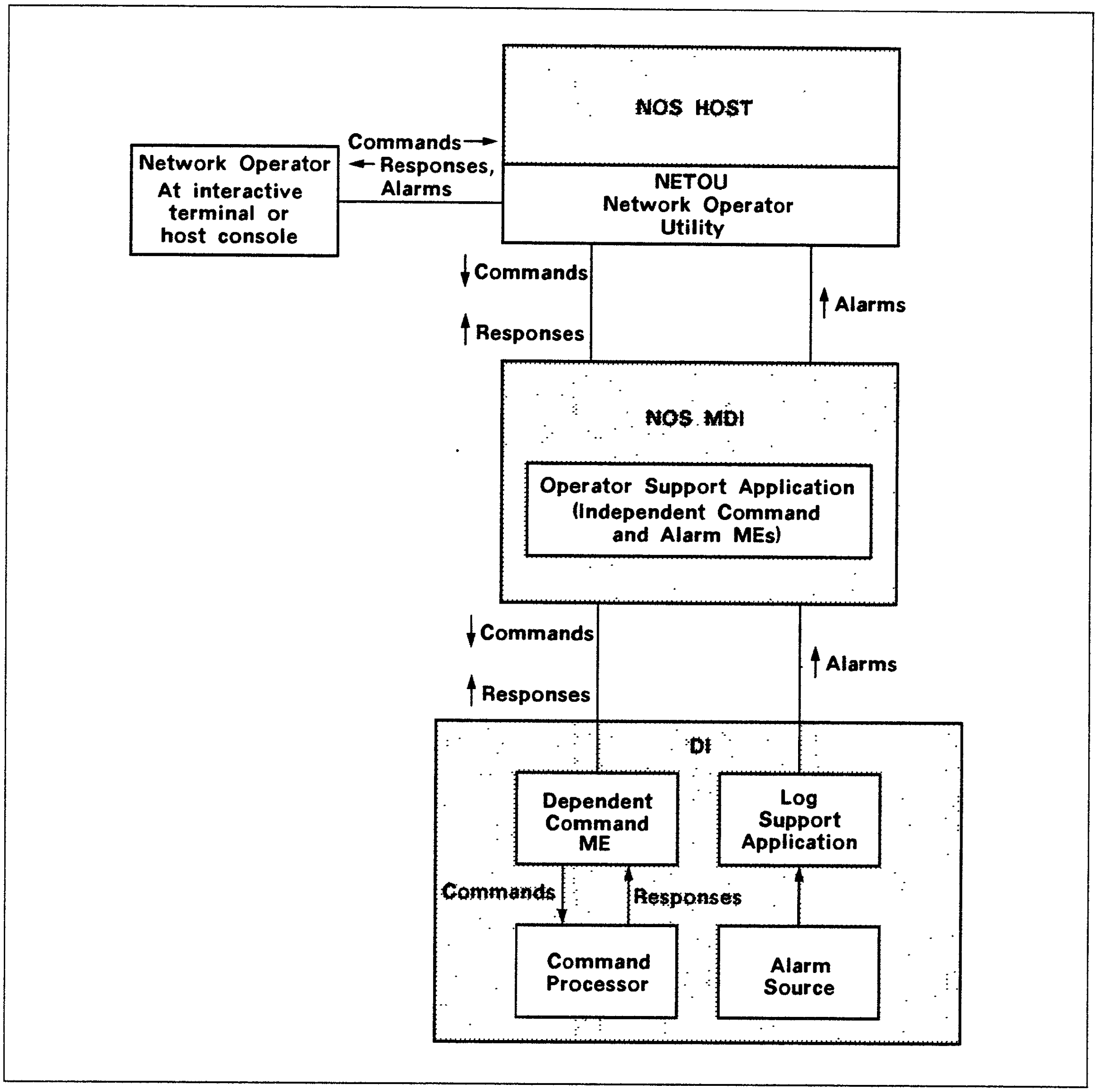


Figure 2-2. NETOU Operating Environment for NOS

Network Operations from an Interactive Terminal

At an interactive terminal, you communicate to CDCNET via NETOU through a normal interactive terminal connection.

NETOU Terminal Display Format

At a terminal, NETOU uses virtual line mode (as opposed to full screen mode) for display output. Commands are entered on a line-by-line basis. Command responses are returned in a line-by-line format as well. You will use some utilities to perform network operations tasks that use full screen mode. These utilities run outside of NETOU, and include NPA, Network Logfile Termination Utility (NLTERM), and MANCC.

Login

Before entering any NETOU commands, you must create a connection to the host system using the CREATE_CONNECTION (CREC) terminal user command, as in the following example in which the user creates a connection with the host system by specifying the title NOS100.

```
create_connection,nos100
```

If you need to review how to use the CREATE_CONNECTION command, refer to the CDCNET Terminal Interface Usage manual.

To enter NETOU, log into NOS and select the NETOU application. Enter your family, user name, password and NETOU, separating each by commas (if you use the default family name, log in beginning with a comma).

family,user name,password,netou

Example:

```
nosfam,bss,sunra,netou
```

If you are validated to access NETOU, you will be connected to NETOU and you will see the following message.

```
WELCOME TO NETWORK OPERATOR UTILITY
CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1987.
```

If you are not validated to access the NETOU application, you will receive the following response.

```
INVALID APPLICATION, TRY AGAIN
```

If you get this message, ask the network administrator at your site if you have been validated to use NETOU.

You may optionally want to create a file containing commands to be executed every time you log in. This NOS indirect access file NETOPRP resides in the operator's catalog. Typically, this file defines your command environment. For more information on prologs, refer to Using a Prolog in the Session Control on NOS section of chapter 3.

Selecting an MDI or MTI

NOTE

This section is provided for site configurations that have more than one MDI connected to a host. Two MDIs that share only a single NOS host are considered separate catenets.

When you log in to NETOU, your job's connection is switched to NETOU. NETOU responds by connecting your operations station to the default MDI or MTI to receive your network commands and route them through the network. If there is more than one MDI or MTI available for you to select for an operations session, NETOU responds in one of two ways.

- NETOU automatically selects an MDI/MTI for you.
- NETOU prompts you to select an MDI or MTI.

You must also select an MDI if the currently-selected MDI breaks its connection with NETOU.

Until an MDI becomes available and you select one, you may only enter the following commands.

DISPLAY_CONNECTED_MDI
DISPLAY_ALARM_HISTORY
ROUTE_ALARM
ROUTE_COMMAND_RESPONSE
SET_COMMAND_MDI
DISPLAY_COMMAND_LIST
DISPLAY_COMMAND_LIST_ENTRY
HELP
DISPLAY_COMMAND_INFORMATION
QUIT
LOGOUT
BYE
GOODBYE
HELLO
LOGIN

All other commands are ignored.

If more than one MDI or MTI is connected to NETOU, you will receive a message listing all the MDIs which you can select to connect with the network. The display you receive will depend on the number of MDIs and/or MTIs defined at your site. The following is an example of such a message:

```
  STATUS OF CONNECTED MDIs
  NODE   CURRENT     SYSTEM
 NUMBER   STATE      TITLE

  043    AVAILABLE   MDI_8A
  044    AVAILABLE   MDI_85
```

If more than one MDI has established a connection with NETOU, as in the example above, you will also receive the following message.

```
More than one MDI available.
Please select an MDI by the following command:
SETCM [MDI=<name>]
Parameter is optional, if omitted,
then default MDI = <default MDI title>
```

The command you enter is called SET_COMMAND_MDI (SETCM). The value of <MDI title> is the MDI to which you will be connected if you do not specify an MDI. If you enter the SETCM command with no parameter, the first available MDI in the list in the AVAILABLE state, is selected.

A default MDI can be defined in the job statement for NETOU in a host file NAMSTRT. If the connection with this default MDI is broken, NETOU will reselect the default MDI. Unless there is more than one MDI at your site, or if you plan to switch between MDIs, you can use the default MDI. For more information on how to select a default MDI refer to the NOS Version 2 Installation Handbook.

Once you have selected an MDI for communication with the network, you will receive that MDI's title in a message sent from that MDI. If you need to check which MDI or MTI you have currently selected, enter the DISPLAY_CONNECTED_MDI (DISCM) command.

NOTE

You will receive alarms that are sent through the selected MDI or MTI. If alarms from more than one catenet are desired, you must execute multiple SETCM commands.

Using a Prolog

Refer to Using a Prolog in chapter 3, if you wish to create a file containing a series of commands that you execute every time you establish a connection.

Prompts

You immediately get the following prompt after logging in to NETOU.

```
NOU/
```

This prompt indicates that you are logged in to NETOU and can begin entering NETOU commands. NOU/ is displayed as a prompt until you select another application, such as IAF.

You immediately get the following prompt after entering the SENCS command.

```
SENCS/
```

This prompt indicates that you are in the SEND_COMMAND_SEQUENCE mode. SENCS mode allows you to send one or more commands to the same system(s) without enclosing the command within a SENC command. The commands you enter following this prompt are sent only to the systems listed in the system parameter of the SEND_COMMAND_SEQUENCE command. SENCS displays as a prompt until your enter ** to exit the SENCS mode.

Paging

Paging allows you to move forward within a display on the terminal screen. You may enable or disable paging using the CDCNET terminal command called CHANGE_TERMINAL_ATTRIBUTES (CHATA). To do this, enter the network command character (NCC) (shown here as a percent sign (%), but the actual NCC may differ for your terminal), and the CHANGE_TERMINAL_ATTRIBUTES command, as in the following example.

```
%CHANGE_TERMINAL_ATTRIBUTES HOLD_PAGE=ON or OFF
```

ON enables paging; OFF disables paging. The default is for paging to be OFF. When paging is ON, to scroll to the next page of text, enter a carriage return or a control character. For more information about the CHANGE_TERMINAL_ATTRIBUTES command and the network command character, refer to the CDCNET Terminal Interface manual. Instructions for paging at the K display console are provided later in this section.

Displaying Job Status Information

Displaying job status allows you to monitor the progress of your job through the CDCNET network.To do this, enter the network command character (NCC) shown here as a percent sign (%) followed by an e. The actual NCC may differ at your site. The first two lines will tell you the current routing of the command responses and alarms. The third line will tell you that you are in SENCS mode that is, if you are in the SENCS mode. A list of the DIs to which th commands are being sent in SENCS mode follows. The last line will tell you the current status of your job.

```
%e

Command responses routed to DISPLAY.
Alarms routed to DISPLAY.
You are currently in SEND_COMMAND_SEQUENCE mode
Commands sent to <list of DIs>
You may enter commands.
```

Logout

When you want to log out from the NETOU application (and optionally log in to another application such as IAF), enter one of the following commands.

HELLO,application
BYE
LOGOUT
LOGIN,application
GOODBYE
QUIT

Examples:

The following example logs an operator out of the NETOU application and selects the IAF application.

```
hello,iaf
```

The following example logs an operator out of the current NETOU session and begins a new NETOU session.

```
hello,netou
```

Network Operations from a NOS Host Console

At a NOS host computer console (a CC545 console or a 721 terminal), your interface to CDCNET is through the standard Network Access Method (NAM) host operator · interface, the NAM K display. This section focuses on using the NAM K display to access and use NETOU. For background information on host console operations and K displays, refer to the NOS 2 Analysis handbook.

NETOU K-Display Format

The K-display format used during the NETOU application is identical to the standard NAM K display used for NOS Operations. For further information about K displays, see the NOS Operations handbook. Figure 2-3 shows a typical K display used for CDCNET network operations.

```
K,NAM.
13:30:45 86.01.10
           MID=81   NOS43C/14R8117KD
..................................................................

    NETWORK_OPERATOR_UTILITY     86/11/10    13.30.45    1478
   WELCOME TO NETWORK OPERATOR UTILITY
    CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1986.
             :

        (data area -- 31 display lines maximum)


READY..     (message line)

ALERTS  (alert line -- a list of applications requesting your attention)

NETOU    SETCM,MDI_80
```

Figure 2-3. NETOU K-Display Format

The NETOU in the lower left corner of the operator entry line indicates that you are logged in to NETOU. To the right of NETOU you will see the last command entered. This field contains 40 characters or less. Commands longer than 40 characters are not completely displayed. NETOU uses the K-display alert line and the operator entry line similarly to the standard NAM applications. NETOU does not use the host message line.

The K display has two data areas, left and right. The left data area displays commands, responses, network alarms, and operator prompts. You may display data on this side as a continuous scroll or view it page-by-page. Refer to the discussion on paging of the K-display, later in this chapter. The right data area is not used for NETOU operations in this release.

Login

Use the following procedure to log in to NOS and select NETOU from a host console.

1. To access the NAM K display from the host console, enter the following.

```
K,NAM.
```

2. Select NETOU:

```
K.AP=NETOU
```

3. The NETOU application responds by clearing the left data area and sending the following prompt.

```
READY..
PLEASE ENTER *USERNAME,PASSWORD*,
ENTER VALUES IN ONE LINE, SEPARATED BY COMMAS.
READY..
```

Enter your user name and password.

```
user name,password
```

Your user name must be a member of the operating system's default family. For a valid login (a login that is known to the operating system and authorized for CDCNET control access), NETOU responds by sending the following message,

```
USER VALIDATION SUCCESSFUL,UN=<user_name>
```

and then connects your session to the default MDI or MTI to receive your network commands and route them through the network. If there is more thanone MDI or MTI available for connecting to the network, you will be prompted (if your site selected the prompting option)to select the MDI or MTI to be used for your operations session. If login is invalid, NETOU reissues the prompt for a valid login. See NOS Version 2 Installation Handbook for more information on selecting the prompting option.

4. You will receive the status of connected MDIs at your site, and a prompt to choose an MDI, if more than one MDI is available.

```
  STATUS OF CONNECTED MDIs
  NODE    CURRENT     SYSTEM
NUMBER     STATE      TITLE

  043     AVAILABLE   MDI_8A
  044     AVAILABLE   MDI_85
```

If more than one MDI has established a connection with NETOU, as in the previous example, you will also receive the following message.

```
More than one MDI available.
Please select an MDI by the following command:
SETCM [MDI=<name>]
Parameter is optional, if omitted,
then default MDI = <default MDI title>
```

5. Enter SET_COMMAND_MDI (SETCM). The value of <default MDI title> is the default MDI to which you will be connected if you do not specifically select an MDI. The default MDI is the first MDI listed. If you enter only SETCM with no parameter, then the first DI in the list is selected.

   You will receive a message showing the current user name in effect when the K display is reassigned after you have logged in.

   ```
   YOU ARE CURRENTLY LOGGED IN AS UN=user_name
   ```

   You may also see alarms that have been sent since a DISPLAY_ALARM_HISTORY command was issued, and a notification of the current operator state, such as command in progress.

You may wish to create a prolog, a file containing a series of commands that you execute every time you establish a connection. For information on how to create a prolog, refer to Using a Prolog, in chapter 3.

Logout

To log out from NETOU, enter any of the following logout commands:

K.LOGIN
K.LOGOUT
K.GOODBYE
K.BYE
K.QUIT
K.HELLO

All the above logout commands perform two actions: they terminate the current session and begin a new session.

After logout, a login prompt will be displayed. You must type K.* to return the K display to NAM control. Once you log out, alarms issued by the network are discarded. Any commands you sent prior to the logout may or may not complete, but you will not receive responses to these commands.

Exiting and Resuming NETOU Sessions

At the host console, you may exit NETOU without logging out of NETOU completely. To do this, enter:

K.*

K.* returns the K display to NAM control. NETOU remains active and retains your login. NETOU continues to monitor the network for alarms, even though you are not currently using the application. If new alarms occur during this time, the following message appears on the K display alert line.

```
NETOU
```

Any alarms received at your operations site can be displayed after you resume using NETOU.

To resume your operations session, enter:

```
K.AP=NETOU
```

NETOU returns the following message.

```
YOU ARE CURRENTLY LOGGED IN AS UN = <username>
```

Immediately following the above message, the most recent alarms will automatically be displayed. Most recent alarms are those that have been sent since the last DISPLAY_ALARMH_ISTORY command was entered. After alarms are displayed, you will receive status information (refer to Displaying Job Status Information, in this chapter), followed by either the READY.. prompt or information that a command is in process. The NETOU prompt will be cleared from the information alert line.

Prompts

Common prompts at the K display include the following.

Prompt	Description
READY..	Command entry is allowed.
MORE DATA..	Page wait is on and more pages (screens) of data exist. Enter K.+ to see the next page of data or K.- to turn page wait off and see the rest of the data.
REPEAT..	You entered a command before the NETOU application was ready to receive it. Wait until you see the READY.. prompt, and reenter the command.
COMMAND TOO LONG	The command you are entering is too long for the K display (see Continuing Commands Over Several Lines).
LINE TOO LONG	The line of data you are entering is too long for the K display (see Continuing Commands Over Several Lines).

Most prompts are displayed at the bottom left corner of the K display's left data area. The REPEAT.. prompt is displayed at the right margin of the operator entry line.

Paging

Some command responses fill more than one screen of the K display. When page wait is on, the MORE DATA.. prompt indicates this. You may view additional screens of data (also known as paging) and control paging of the data areas by entering the following commands. You may enter commands to turn paging on and off for the left data area. By default, paging is off at the K display.

Command	Description
K.+	Turns paging on for left data area. When you first enter NETOU mode, paging for the left data area is off. Once paging is on, NETOU will only display one page at a time of a multi-page response. Multi-page responses are indicated by the MORE DATA.. prompt. You may scroll to the next page by again entering K.+.
K.-	Turns paging off for the left data area. If you enter K.- instead of K.+ when the MORE DATA.. prompt appears, paging is shut off. The screen immediately displays all responses, and MORE DATA.. is not displayed again.

If you change page wait from off to on or on to off, the success response is as follows:

```
PAGE ACCEPTED.
```

K-Display Console Entry Restrictions

All commands at the K display are entered as follows:

K.command

The K. prefix is required. The syntax used for the command portion is the same as that used at an interactive terminal. Refer to the Common Network Operations Features section in this chapter for more information on command syntax.

Normally, once the K display is active, the K. is automatically generated each time you enter a command. If you cancel the automatic feature by pressing the erase (left blank) key on the system console, you can restart the automatic process again by reentering the K. before the next command. Enter a carriage return to indicate the end of a command.

Entering Characters Not Supported at a NOS Host Console

NETOU commands use a subset of the syntax for NOS/VE SCL commands. SCL uses the ASCII character set, which has characters the NOS host console (CC545 and 721) does not support. On the NOS host console, you must type two characters, or an escape sequence, to designate the ASCII characters not supported on the console.

On the NOS host console screen, unsupported ASCII characters are designated by other characters. For a character which represents more than one ASCII character when displayed, such as the asterisk (*), the only way to know which ASCII character it represents is by the display's context. Table 2-1 shows escape sequences for unsupported ASCII characters and how these characters are represented on the console screen.

The following example compares command entries made at a terminal that supports the full ASCII character set with the same entries made at a NOS host console using the escape sequences. In this example, the hyphen is used rather than the /0 sequence to represent the underscore character.

ASCII terminal entry:

```
send_command command='display_hardware_status',system=north_tdi_1
```

System display console entry:

```
SEND-COMMAND COMMAND=/*DISPLAY-HARDWARE-STATUS/*,SYSTEM=NORTH-TDI-1
```

Table 2-1. NOS Host Console Escape Sequences and Displays

Character	Name	Escape Sequence On Keyboard	Displayed On Screen As:
^	Circumflex	/1	/1
"	Quotation Marks	/2	/2
#	Number Sign	/3	/3
$	Dollar Sign	/4	/4
@	Commercial At	/5	/5
;	Semicolon	/6	/6
?	Question Mark	/7	/7
{	Opening Brace	/8	/8
}	Closing Brace	/9	/9
_	Underline	Hyphen (-) or /0	-
[	Opening Bracket	/(	/(
]	Closing Bracket	/)	/)
>	Greater Than	/+	/+
<	Less Than	/=	/=
'	Apostrophe	/*	/*
/	Slant	//	/
!	Exclamation Point	None	.
%	Percent Sign	None	*
&	Ampersand	None	+
\	Reverse Slant	None	*
^	Grave Accent	None	*
|	Vertical Line	None	*
~	Tilde	None	*
:	Colon	/,	.
-	Minus, Hyphen	/-	-
a..z	Lowercase	/A../Z	A..Z

Continuing Commands

The K display does not accept input of more than 50 characters after the K. If you enter a command that goes over this limit, you will receive one of the following prompts in the lower left corner of the console screen.

```
LINE TOO LONG

COMMAND TOO LONG
```

When this happens, the command entry is not processed. You may not enter anything else until you clear the entry by one of the following methods.

- Press the backspace key repeatedly, until you have fewer than 50 characters.
- Erase the entry by using the left blank (erase) key on the system console keyboard. Then reenter the command starting with the K.

To enter command strings that are longer than 50 characters, use the continuation symbol, the ellipsis (..), before you enter the 48th character, and enter a carriage return. Continue the command on the next line. The following examples show how to enter a multiple-line command from a system console. Assume that each line ends with a carriage return.

```
K.SEND-COMMAND ..
K.C=/*DISPLAY-LINE-STATUS LINE-NAME=(COMPSCI-02 ..
K.ENGINEERING-PORT-1 ENGINEERING-PORT-2 ..
K.ENGINEERING-PORT-3) SYSTEM=NORTH-TDI-2/*
```

Command Syntax for NOS NETOU

This section describes the special syntax rules and the process used for sending CDCNET network operations commands from your operations station to the network in a NOS-based operations environment.

In a NOS environment, NETOU has the following types of commands.

- Commands executed on the host (session control commands).
- Commands executed in the MDI through which you are communicating with the network (session control commands).
- Commands executed in DIs throughout the network (network commands).

All these commands follow a subset of the NOS/VE SCL syntax (refer to command syntax in the Common Network Operations Features section of this chapter). All network operations commands share the following properties in a NOS environment.

- Lowercase letters are interpreted as uppercase letters, with the exception of lowercase strings enclosed within single quotation marks (').
- Entering more than one network operations command per entry line is prohibited.

Some commands require parameters, such as FILE_NAME, that are passed on to NOS. The values allowed for these parameters have the same syntax and limits as those used in the NOS command language.

Entering NETOU Commands on NOS

NETOU commands are valid only within a NETOU session. The session begins when you select NETOU. The session ends when you log out of NETOU.

SEND_COMMAND (NOS Version)

To send network commands through the network, you use SEND_COMMAND. (SENC), transmitting the network commands to the DI you specify. Except for the session control commands described later in this chapter, you must embed all network commands in a SEND_COMMAND. To use this command, enter:

SEND_COMMAND COMMAND=string,SYSTEM=name

COMMAND (C) is the network command to be sent to the DI specified with the SYSTEM parameter. Enter the command as a string value enclosed by apostrophes (').

NOTE

If the network command you are sending contains any apostrophes, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized. Otherwise, NETOU will assume the embedded apostrophe signals the end of the network command, and errors will result.

SYSTEM is the logical or physical DI name or list of DI names to which you want to send the command. SYSTEM is an optional parameter. If you omit this parameter, the last DI to which you sent a command is used. If SYSTEM is omitted on the first SEND_COMMAND you use after you log in to NETOU, the selected MDI is used as the value for the SYSTEM parameter. The SYSTEM parameter may specify a maximum of 15 systems to which you want to send a network command with a single SENC command. If a network command is sent to more than one DI, a response from each DI must be received for the command to complete. The SYSTEM parameter is optional for SEND_COMMAND in NOS environments, but required for SEND_COMMAND in NOS/VE environments.

NOS SEND_COMMAND Examples

1. The following command sequence would be entered to stop traffic on a communications line connected to a DI named TDI_3.

   ```
   send_command command='stop_line line_name=line3',system=tdi_3
   ```

   The actual command to stop communications traffic is enclosed within a SEND_COMMAND that specifies the DI (TDI_3) to which the line is connected.

2. This SEND_COMMAND command is used to send a DISPLAY_LINE_STATUS command sent to the same DI as in example 1 (TDI_3). In this example, the SYSTEM parameter can be omitted, since the previous SEND_COMMAND specified TDI_3.

   ```
   send_command command='display_line_status'
   ```

Common Network Operations Features

This section describes features of NETOU that are common to both NOS/VE and NOS operations environments. The following features are described: Command syntax rules, descriptions of common command verbs, wildcard characters (NOS/VE only for this release), order of command execution, command responses, alarms, and severity levels for responses and alarms.

Command Syntax

This section outlines the syntax rules for the CDCNET commands described in this manual. All commands follow a subset of the NOS/VE SCL syntax. This section is provided to give you sufficient information to understand the commands used in this manual. For more information on SCL command syntax, refer to the SCL for NOS/VE Language Definition manual. Commands used in a NOS environment have additional properties (see Command Syntax for NOS NETOU).

Command Format

A command is in the following form.

command_name parameter_1=value_1,parameter_2=value_2,...

Example:

```
DISPLAY_HARDWARE_STATUS DEVICE_NAME=$LIM0_PORT0,DISPLAY_OPTION=EXPANDED
```

Either a blank or a comma can be used as a separator. The underscore character cannot be omitted. Command strings may be up to 256 characters long. The maximum size of a SEND_COMMAND command (SEND_COMMAND plus command to be sent) is 512 characters. You may continue entering a command on another entry line using the ellipsis (..), as shown in the following example.

```
senc c='start_process_metrics p=xns_transport,..
g=(summary,expanded)',s=mdi_3
```

Command Abbreviations

Command names are abbreviated by taking the first three characters from the verb portion of the command name and combining them with the first character from the remaining words in the command. The abbreviated form of a command name having a plural form is the same as the abbreviation for the singular form.

For example, DISPLAY_HARDWARE_STATUS is abbreviated by taking DIS from DISPLAY and combining it with the H from HARDWARE and the S from STATUS to form.

DISHS

Parameter Abbreviations

Parameter names are abbreviated by taking the first character from each word in the parameter name. For example, the parameter LINE_NAME has the abbreviated form LN.

Parameters and Parameter Values

Parameters consist of a parameter name followed by an equal sign and a parameter value. A parameter value may be a list of values, as in:

parameter=(value_1,value_2,...)

or a list of lists, as in:

parameter=((value_1,value_2),value_3,(value_4,value_5...)...)

The following types of parameter values are allowed: string, name, integer, boolean, and keyword value.

A string is any sequence of ASCII characters enclosed by apostrophes ('). Most of the network operations commands must be entered as a string value within SEND_COMMAND. The enclosed command string must be surrounded by apostrophes. If you include an apostrophe within a string value, you must use two consecutive apostrophes for the embedded apostrophe character to be recognized, as in the following example.

```
send_command c='write_terminal_message,..
m=(''New communications configuration tomorrow'',''Network down ..
until 10:00.'')',s=tdi1
```

An SCL name is a combination of from 1 through 31 alphabetic characters (ASCII characters A through Z and a through z), digits (ASCII characters 0 through 9), and/or special characters (underline [_], dollar sign [$], number sign [#] and commercial at [@]). Lowercase is folded to uppercase in a name.

An integer parameter value represents a binary, octal, decimal, or hexadecimal integer value. Integer values may be expressed as a combination of digits or for hexadecimal integers, A through F (uppercase or lowercase). A hexadecimal integer must begin with a digit. SCL makes no distinction between uppercase and lowercase characters in hexadecimal integer constants.

Integer parameter values may be expressed as: integer (radix), followed by a range of integer values. If you do not specify a radix, the decimal system (base 10) is assumed. Any radix between 2 and 16 is accepted. A radix must be surrounded by opening and closing parentheses, as in 1FFFF(16) and 101(8).

NOTE

When you specify a radix, be sure to type the radix correctly.

In command descriptions, when two integers are separated by an ellipsis (..), a range of integer values is possible. The allowed value may be the first value through the second value. No spaces are allowed around the ellipsis. For example, the parameter value BUFFER_SIZE = 64..4096 indicates that any value from 64 through 4096 is possible for the BUFFER_SIZE parameter.

A boolean parameter value represents a condition of either TRUE or FALSE. There are three possible words used for both TRUE and FALSE conditions. For a TRUE condition, you may specify TRUE, YES, or ON. For a FALSE condition, you may specify FALSE, NO, or OFF. NOS NETOU only supports YES and NO.

A keyword value is a parameter value that has a special meaning in the context of a particular parameter. For example, the command DEFINE_LINE has a parameter LINE_TYPE, where two types of lines, switched and dedicated, are allowed. Two keyword values are allowed for this parameter: SWITCHED and DEDICATED. You specify one or the other by providing the appropriate keyword value for the parameter. The keyword value ALL is frequently used in commands to select all available options for a parameter value.

Default Parameter Values

Not all parameters require you to provide values. In the command descriptions, required and optional parameters are designated. Most parameters have a value called a default parameter value that is provided if you do not specify the parameter with the command. Default parameter values are specified in command descriptions.

Command Entry

You can enter the commands in this manual in two ways.

- Position-dependent
- Position-independent

In position-dependent format, you supply values for parameters in the order specified in the command format, without entering parameter names or equal signs. Separate parameter values with commas. If you omit any parameters, you must supply a comma for the missing parameter.

In position-independent format, you supply the values for the parameters by specifying the parameter name and the equal sign before the value for each parameter. You can enter the parameters in any order.

Command Verbs

This section explains the verbs used in several common network operations command types. Commands beginning with these words comprise the bulk of network operations commands. For the complete list of network operations commands and command descriptions, see the Network Commands chapter in part 2 of this manual.

Cancel

Cancel commands delete the logical configuration of the element you specify. For example, you may cancel the logical configuration of an Ethernet network solution using the CANCEL_ETHER_NET command. The network solution's logical configuration is deleted. If you want the network solution to support data transfer again, you must redefine the network using a DEFINE command type, described below.

Change

Change commands change the current logical configuration of a hardware component, the values of certain aspects of a DI's operating system such as buffer size and memory management, or the set-up of the network's system for reporting alarms. These changes may be made while the network is operational; you don't have to shut down and reload DIs to use change commands.

Define

Define commands create a logical configuration of the element you specify in the network. Define commands are a part of the set of configuration commands, and are used in DI configuration files. These commands are also used if you cancel a component's logical configuration and want to redefine it.

Display

Display commands return information you request to your operations terminal or console screen. There are display commands to display the following information.

- Status for hardware and software elements of a DI.
- Configuration parameters for network elements.
- The list of log messages and alarms to be transmitted from a DI.
- The current date and time registered at a specific DI.
- Diagnostic test results.

For commands that display several parameters, you can select which parameters you want displayed. These commands have a parameter called DISPLAY_OPTION (DO), which allows you to specify only parameters that are of interest to you. You may choose one, several, or all of the options that a DISPLAY_OPTION parameter allows.

For example, the DISPLAY_SYSTEM_OPTIONS (DISSO) command, which displays the current value of DI system program attributes, has a DISPLAY_OPTION parameter. DISPLAY_OPTION allows you to choose from among several configuration attributes you want displayed, by specifying keyword values such as DATA_BUFFER_SIZE, BUFFER_PERCENTAGE, MEMORY_MANAGER_PERIOD, and CLOCKING_SYSTEM.

Start

Start commands begin the specified action, or enable the specified component to begin data communications. Some start commands make an element you specify operational, or ready for data transfer. For example, you may start communications traffic on a communication line from a LIM to a terminal using the command START_LINE. Other start commands begin online diagnostic tests (such as START_CIM_TEST and START_ESCI_TEST), and statistics collection (such as START_LINE_METRICS and START_NETWORK_METRICS).

Stop

Stop commands end the specified action, or disable the specified component from performing data communications. Some stop commands stop the support of data transfer on the network element you specify, such as STOP_LINE and STOP_NETWORK. Other stop commands stop diagnostics, and statistics collection.

Order of Command Execution

Commands sent to a DI are executed in the order received. There is no underlying priority as to which command is executed first. Commands from operators at different stations that affect overlapping sets of DIs may be received in a different order at each DI. If there is more than one CDCNET network operator currently logged in and sending commands, there is no guarantee that commands sent from one network operator to network components will be performed in sequence before those sent from another network operator.

Command Responses

All commands entered generate a response. This section describes command responses.

Command Response Format

CDCNET command responses have the following format (brackets indicate optional portions of the response).

```
FROM tttttttttttttttttttttttttttttttt  ccccc
[--ssssssssssss-- ]response text
```

ttttttttttt...	Logical or physical name of system sending response.
ccccc	Numerical identifier for the command response. NOS/VE does not display this identifier for informative command responses; NOS does. Using the message number, you can reference the command response's description in the CDCNET Diagnostic Messages manual.
ssssssssss	Severity level of command response (see Severity Levels for Command Responses and Alarms, later in this chapter) This severity level is not displayed for every response. If no severity level is displayed, the response is informative and the command has completed successfully.
response text	The response text may either directly follow the severity level or begin on the next line.

For NOS/VE environments, a normal CDCNET command response is written to the output file when it includes response text. An abnormal CDCNET command response is always written to the standard file $RESPONSE.

The following is an example command entry and command response (NOS host). It shows a command called DISPLAY_HARDWARE_STATUS (DISHS) being sent to a DI, and the response sent back to the network operator.

Command:

```
senc s=mdi_1,c='display_hardware_status'
```

Command response:

```
FROM MDI_1                                   33021
Hardware Status
device name    status    state         version   lim/bank/port   type
$MPBO          on        active        0000
$PMM1          on        active        0008
$SMM2          on        active        0001      2
    3          off
$CIM4          on        configured    0001      0,1,2,3
$CIM5          down      not config.   0001
$ESCI6         on        active        0000
$MCI7          on        active        0000
$LIM0          on        enabled                 4               RS232
$LIM1          down      configured              4               RS232
$LIM2          on        enabled                 2               RS449
$LIM3          on        not config.             2               RS449
```

The following is an example command entry and command response (NOS/VE host). It shows a command called DISPLAY_HARDWARE_STATUS (DISHS) being sent to a DI, and the response sent back to the network operator.

Command:

```
senc s=mdi_1,c='display_hardware_status'
```

Command response:

```
FROM MDI_1
Hardware Status
device name     status     state         version     lim/bank/port    type
$MPB0           on         active        0000
$PMM1           on         active        0008
$SMM2           on         active        0001        2
    3           off
$CIM4           on         configured    0001        0,1,2,3
$CIM5           down       not config.   0001
$ESCI6          on         active        0000
$MCI7           on         active        0000
$LIM0           on         enabled                   4                RS232
$LIM1           down       configured                4                RS232
$LIM2           on         enabled                   2                RS449
$LIM3           on         not config.               2                RS449
```

Other examples (NOS host):

```
send_command c='display_date_and_time',s=di_sn093

FROM DI_SN093                                    33525
 System date and time
      31/01/85 23:20:24

send_command c='display_date_and_time',s=di_sn093

FROM DI_SN093                                    33525
 System date and time
      31/01/85 23:22:17
```

Other examples (NOS/VE host):

```
send_command c='display_date_and_time',s=di_sn093

FROM DI_SN093
 System date and time
      31/01/85 23:20:24

send_command c='display_date_and_time',s=(di_sn093)

FROM DI_SN093
 System date and time
      31/01/85 23:22:17
```

Some command responses are common to all network commands. For example, responses that indicate that the DI or component cannot be located or is unavailable may occur for any command sent to a DI. Also common to all commands are error responses that indicate unknown commands, invalid parameters and incorrect parameter values. These are called command parser errors. Command parser errors abort execution of commands. These common responses are not documented with the commands chapter 6, 7, and 8. Only responses that are uniquely defined for the command are documented there. All command responses are documented in the CDCNET Diagnostic Messages manual.

Loss of Commands and Responses

Network commands to specific DIs are sent by transport connections that ensure commands are delivered to the correct DI and that loss of commands in transmission cannot occur. However, a destination DI could fail while the command is executing, or the command processor in the DI could stop abnormally. To allow for such events, NETOU times the response for any command and declares a command failed if no response is received from the CDCNET system within 120 seconds after the command is sent. For commands that do not send a response within 120 seconds, the following response is sent.

NOS example:

```
--ERROR-- No response received from system <name> for the CDCNET command
<command_name>.
```

NOS/VE example:

```
--ERROR--No response received from system <name> for the last CDCNET command
```

Break Processing (Response Suppression)

With break processing, you may suppress responses to network commands in progress (keep any output from commands from being displayed on your screen). Commands with suppressed responses will complete, but no response for the commands will be delivered to your operations station.

Command response supression does not abort command processing. You cannot abort commands that are being processed at the destination DIs. Once received at a DI, commands complete regardless of what you enter from your terminal or the host console. When you suppress responses, the next command entry prompt (nou/ on NOS/VE and NOU/ on NOS) indicates the end of response suppression. Commands entered after you receive that prompt execute normally and return responses.

Response Suppression on NOS/VE

On NOS/VE, you initiate response suppression by entering the user_break_1 or a user_break_2 at an interactive terminal. If an included file is executing when response suppression is initiated (see Using Command Files in chapter 3), response suppression both suppresses responses for commands in progress and terminates NETOU processing of the file.

When a user break sequence is entered, NETOU responds with the Terminal Manager response to a user break. The response to the user break also identifies commands for which responses have not been received and commands that have unknown destinations. The following messages are used to indicate these conditions.

```
No response received from system <string> for the last CDCNET command.

System <string> is unknown.

No response received to connect request to system <string>.
```

Response Suppression on NOS

On NOS, when a break command is issued, some commands sent to a DI may still be processed, and others may have the output discarded. You initiate response suppression using one of the following methods.

- At an interactive terminal, enter the user_break_1 or user_break_2 sequence. NETOU will respond with the following message.

  ```
  Pending responses suppressed
  ```

- At a host console, enter K./

You can enter a command response suppression command while a file of network operations commands is being executed (see Using Command Files in chapter 3). Command response suppression both suppresses responses and terminates NETOU processing of the command file.

Alarms

Alarms may be sent from DIs to your operations station during an operations session. These alarms are unsolicited; they are not responses to commands, and you may receive them at any time during an active NETOU session.

On NOS, alarms are always activated. You do not have to enter a command to activate their transmittal to your operations station. In NOS/VE environments, alarms are not initially activated. You must explicitly activate alarms by entering the ACTIVATE_ALARMS command before you can receive alarms. Rather than manually activating alarms every time you begin an active NETOU session on NOS/VE, you can automatically activate alarms through your NETOU prolog by placing the appropriate commands in your prolog. See Session Control on NOS/VE in chapter 3 and the ACTIVATE_ALARMS command in chapter 6 for more information.

Alarms alert you to a wide range of conditions that occur in a network, from the completion of a diagnostic test to the failure of a hardware component. In addition, any messages sent to you from the network's terminal users appear as alarms at your display.

When a DI completes being loaded and logically configured, alarms generated during the logical configuration are sent to your operations station.

Much of your network operations work involves responding to CDCNET alarms.

Alarm Format

CDCNET alarms have the following format (brackets indicate optional portions of the response).

```
****** ALARM FROM  ttttttttttttttttttttttttt  date/time  ccccc
[--sssssssssssssssss-- ]alarm text
```

tttttttttt... Logical or physical name of system sending alarm. An alarm generated by NETOU itself, such as an alarm issued when an MDI connection is broken, will display NETWORK_OPERATOR_UTILITY in this field.

ccccc The numerical identifier for the alarm message. This identifier is displayed for all alarms and is intended to help you index into the CDCNET Diagnostic Messages manual for a description of the message.

sssssssssssss Severity level of the alarm (refer to Severity Levels for Command Responses and Alarms, later in this chapter. This field is suppressed for informative alarms; if no severity level is displayed, the alarm is informative. Informative alarms may indicate the completion of an operation (such as a diagnostic), the recording of information (such as statistics), or convey a message from a terminal user. Informative alarms are not the result of incorrect or incomplete CDCNET operation.

The following is an example of an alarm:

```
****** ALARM FROM DI_SN093                  85/01/31  23.24.31  458
       New maximum recovery rate
       Failure ID = 0013
       Threshold count = 1
       Period in seconds = 2
```

Alarm Output

When alarms are sent to you, they are immediately displayed at your screen unless you specifically route alarms to a file only, using the ROUTE_ALARM command on NOS, or connect the alarm output to a file on NOS/VE (see chapter 3).

Alarms appear in the order received. There is no underlying priority as to which alarm is displayed first. At an interactive terminal, alarms are not displayed while you are entering input. Should an alarm be delivered to your terminal, an alarm bell will ring only at interactive terminals, and only on NOS NETOU. At an interactive terminal or host console, alarms may be interspersed within the responses to commands.

Severity Levels for Command Responses and Alarms

The command responses and alarms you receive are grouped into the following severity levels: Informative, Warning, Error and Fatal.

For command responses, Informative and Warning severity level responses indicate a command completed successfully. Error and Fatal severity level responses are considered error responses. An error response alerts you to command errors. The following are descriptions of these severity levels.

Informative

An Informative-level command response indicates successful command completion. Informative alarms are not the result of incorrect or incomplete CDCNET operation. The severity level for informative responses and alarms is not displayed. If you receive a response or alarm without a severity level displayed, the response or alarm is informative.

Warning

A Warning-level command response indicates that a command completed successfully, but that the command may have some unintended effects. For example, some of the definition parameters for a communications trunk may be changed while the trunk is active. Changing those parameters, however, could disrupt communications over the trunk, unless changes at both ends of the trunk are coordinated. Warning-level responses are sent for redundant commands.

Warning alarms alert you to potential network problems. They indicate that a DI or the CDCNET is approaching an error or fatal condition, such as a lack of system buffers. However, no operation is yet incorrect or incomplete due to the condition. Check the alarm's text to determine what you can do to avoid errors in the network.

Error

An Error-level command response indicates that a command failed due to operator error. An error response may indicate, for example, errors detected in command processing, errors in parameters, such as unknown names, and attempts to execute a command which is not allowed. Error-level responses may also indicate that a connection could not be established to deliver a command to its destination system.

Error level alarms indicate the following: the failure of an operation to complete correctly, with the possibility of being recovered by the DI's software; and the failure of a device connected to the DI, such as the loss of a modem signal or communication line.

Fatal

A Fatal-level command response indicates that a command failed due to device failures or lack of resources to complete the command. For example, if there is not enough memory available on a DI hardware device to execute a command, a Fatal-severity level response would be returned.

Fatal alarms indicate the following: the failure of an operation to complete correctly, without the possibility of being recovered (such as the failure of DI system software); and the failure of tasks in the DI system software. When you receive fatal alarms, it is important to intervene when possible to prevent a system failure.

Session Control 3

Session Control 3

This chapter contains descriptions of NETOU session control commands and procedures. Session control is a term used to describe the set of actions you take to define, change and control the online environment for your CDCNET network operations sessions. Examples of session control include routing command responses and alarms to files, and executing files of CDCNET commands. Session control commands differ from other network operations commands because they are not sent to DIs. They define your operations set-up and are not enclosed within SEND_COMMAND.

The chapter is divided into two sections: Session Control on NOS/VE (for NOS/VE-based operations) and Session Control on NOS (for NOS-based operations). Each section provides instructions for using session control commands when doing session control activities. The session control commands are described in Quick Reference format in chapters 6 and 7.

Session Control on NOS/VE

This section describes how to use commands and functions to control your CDCNET network operations sessions in a NOS/VE environment. In a NOS/VE environment, most of the CDCNET operations session control is done through standard SCL functions, commands and services on NOS/VE. If standard SCL functions, commands or services are used to perform any activities, they are referred to in the text, but not described in detail. You will be referred to the appropriate NOS/VE SCL manual for more information.

Session Control Commands

The commands that are specifically used for NETOU session control on NOS/VE include the following.

Command	Description
ACTIVATE_ALARMS (ACTA)	Initiates receipt of alarms from DIs at your operations station. To receive alarms from DIs at your operations station, this command must be entered any time you enter NETOU.
DEACTIVATE_ALARMS (DEAA)	Terminates receipt of alarms from DIs at your operations station.
QUIT (QUI)	Terminates a NETOU session.

In addition, you can use standard NOS/VE commands such as INCLUDE_FILE and CREATE_FILE_CONNECTION in NETOU session control.

SCL Functions for NETOU Sessions

NETOU provides the following SCL functions to help you perform iterative operations and to use NETOU commands in combination.

$NORMAL_RESPONSE

This function returns a value of TRUE if a normal response was received from the last CDCNET command sent by the SEND_COMMAND command. The command format is:

$NORMAL_RESPONSE_(name)_

where _name_ is the name of the system for which the response is to be checked. This parameter is always optional. If the last CDCNET command was sent to more than one system and the <name> parameter is omitted, then a value of TRUE is returned only if all of the responses were normal.

$RESPONSE_IDENTIFIER

This function returns the command response identifier from the response to the last CDCNET command sent by the SEND_COMMAND command. Response identifiers are integers in the range 33000..65535. The meaning of a specific value is described in the CDCNET Diagnostics Messages Manual. The format for this function is:

$RESPONSE_IDENTIFIER*(name)*

where *name* is the name of the system for which the response is to be checked. This parameter is optional if a command is sent to only one CDCNET system. If the last CDCNET command was sent to more than one system, then the name parameter is required.

$MATCHING_NAMES

This function returns a list of CDCNET system names matching a name pattern. The list of names is assigned to an SCL array variable that is then used as the value for the SEND_COMMAND parameter that sets the destination for a series of CDCNET commands. The name pattern may contain wildcard characters. For this release of CDCNET, wildcards are supported for the $MATCHING_NAMES function only. (Refer to Wildcard Characters in this chapter for more information.)

The format of the function is:

$MATCHING_NAMES(string)

where **string** is a string representing the pattern to be matched. This is a required parameter. Enclose the string value within apostrophe characters. Example:

```
$MATCHING_NAMES('DI_*')
```

Wildcard Characters

Optional wildcard characters allow you to address a command to CDCNET systems using names that match a specific name, as modified by a wildcard character. Names used as the destinations for network commands may be modified by the following wildcard characters.

Character	Description
?	Represents any single character.
*	Represents any string of characters.
[ ]	Represents any one of a set or range of characters collated in the ASCII character set. For example, [3ab4] represents any one of the character set 3, a, b, or 4. The abbreviation [3-6] represents any one of the characters 3, 4, 5, or 6.

SCL Procedures for NETOU Sessions

You can create and use SCL procedures that use the functions described in this section to enhance your NOS/VE NETOU environment. For example, you could create a procedure that uses the $MATCHING_NAMES function to send a command to a set of DIs that match a name modified by wildcard characters. Suggested SCL procedures are listed in appendix C of this manual.

Session Control Activities

This section contains instructions for using NOS/VE-based session control commands and functions to set up and control your operations sessions.

Using a Prolog

A prolog is a file containing a list of commands that are executed each time an activity is initiated. You can create a prolog specifically for your NETOU sessions that will be executed every time you access NETOU. A prolog is not required for a successful invocation of NETOU. The commands you put in the prolog are up to you. Any NETOU or NOS/VE commands may appear in the file. For example, the ACTIVATE_ALARMS command must be entered any time you invoke NETOU if you want to enable alarm reporting at your operations station. Instead of entering this command every time, you could put it in your prolog to automatically enable alarm reporting whenever you invoke NETOU.

The default prolog file name is $USER.NETWORK_OPERATOR_PROLOG. However, you may define alternate prologs and put them in any catalog you can access through a normal NOS/VE file reference. When you invoke NETOU with the NETOU command, use the PROLOG parameter to specify the file reference for your prolog.

During NETOU sessions, other files called command files can be used to simplify command entry. The next section provides information on command files.

Using Command Files

Command files contain CDCNET network operations commands (both session and network control commands) as well as any other NOS/VE commands. You can use the NOS/VE command INCLUDE_FILE to process a command file. The INCLUDE_FILE command causes the text of a file to logically replace the occurrence of the INCLUDE_FILE command. The commands in the specified file are then processed. Each line of the command file is executed as if it were an individual command you typed in at your operations site. For more information on the INCLUDE_FILE command, refer to chapter 7 of the SCL for NOS/VE Language Definition manual. You may build command files to perform session and network control activities. A break sequence will terminate command file processing.

Command files can be an efficient way to send commands and save keyboard entry, since you can group several commands that perform a single activity together in a file. Once a command file is created and saved, when you need to perform an activity such as redefining a line, you specify the file with the INCLUDE_FILE command rather than entering all the commands individually. The Network Control chapter describes network operations activities and the commands that perform the activities. You can build command files to perform the activities described there.

You can also use command files to send a command to several DIs. The command file would have the same command on every line, but the DI name specified on the SEND_COMMAND would differ for each line.

Writing Command Files

The following procedure makes use of the concepts for managing NOS/VE files. For more information, refer to chapter 3 of the SCL for NOS/VE System Interface manual. This procedure also assumes you can use the Full Screen Editor (FSE) for NOS/VE.

1. Create and edit a file using the Full Screen Editor by entering the EDIT_FILE command. When creating the file, you must specify the FILE_CONTENT and FILE_PROCESSOR parameters. The FILE_CONTENT = LEGIBLE parameter permits the file to contain character data. The FILE_PROCESSOR = SCL specifies that SCL will process the data. You may put any NOS/VE session control commands and CDCNET network control commands in the file. For network control commands, be sure to enclose the commands within the SEND_COMMAND command. You can also enter other NOS/VE commands in the file. To add comments in the command file, enclose the comment text in quotation marks. If a command and/or its comments continues over several lines, use the continuation symbol (..) at the end of each line.

2. Note the command file's purpose, either in the file itself (as a comment) or in your records. This is important if you have many command files or several versions of a command file.

3. Test the file by attempting to execute it using the INCLUDE_FILE command.

Executing Command Files

To execute a command file, use the INCLUDE_FILE command and specify the file name.

INCLUDE_FILE FILE=file PROMPT=string STATUS=status variable

Only the FILE parameter is required. The FILE parameter specifies the file containing commands to be included. Provide the name of the command file you want to execute. For descriptions of the other parameters, refer to the INCLUDE_FILE description in the SCL Language Definition manual.

The following example shows how command files can be used to send the same command (DISPLAY_DI_SYSTEM_STATUS or DISDSS) to several DIs. Rather than having the operator enter the commands individually, the commands in file DI_STATUS are executed. Comments in the file are enclosed within quotation marks.

```
"File DI_STATUS contains the DISPLAY_DI_SYSTEM_STATUS command."
"When the file is executed, the status command will be sent to"
"the three DIs specified in the file by the SEND_COMMAND."

senc c='disdss',s=mdi1
senc c='disdss',s=tdi1
senc c='disdss',s=tdi2
```

To execute DI_STATUS, the following command is entered.

```
INCLUDE_FILE FILE=DI_STATUS
```

The commands in the DI_STATUS file are sent to the appropriate DIs, where they are executed.

The following command file, DEFINE_ETHERNET, is a standard set of commands used to redefine an Ethernet network solution. Parameter values are left blank so the file can be copied and parameter values can be specified.

```
"File DEFINE_ETHERNET"
"This file is a template file of network operations commands"
"that can be copied and used to define an Ethernet network solution."
"Insert the appropriate parameter values where indicated."
"Not all optional parameters are shown.  If other parameters are added"
"to the command being sent, they must be placed within the final"
"apostrophe character."

Send_command Command='Stop_network Network_name=      'System=

Send_command Command='Cancel_ether_net Network_name=  ',System=

Send_command Command='Define_ether_trunk Slot=  ,Trunk_name=',System=

Send_command Command='Define_ether_net,..
  Trunk_name=      ,Network_ID=       Network_name=       ',System=

Send_command Command='Start_Network Network_name=      ',System=
```

A command file is useful in this situation because defining and starting a network solution involves defining and starting the network at two places. Once a file of commands to define and start a network solution is created, the file can be duplicated and used to define and start the network solution on each DI affected by the definition change. This command file includes comments that describe the file's use.

Activating and Deactivating Alarms

Every DI generates alarms which range from informative messages to indications of software failures (see Alarms in chapter 2). By default, these alarms are not sent to your operations station unless you explicitly activate them. To activate alarms so that they are displayed at your operations station, transmittal from the host to your station must be activated any time you invoke NETOU, by entering the ACTIVATE_ALARMS command. To ensure that this command is entered every time you invoke NETOU, include ACTIVATE_ALARMS in your user prolog (refer to Using a Prolog in this section). Then, when you enter the NETOU command to invoke NETOU, alarms will be activated.

Alarms are deactivated by shutting off the transmittal of alarms from the host to your operations station. To do this, enter the DEACTIVATE_ALARMS command.

For NOS/VE CDCNET operator environments, all alarms received at the operations station are displayed when alarms are activated. Either all DIs in the network send alarms to you, or no DIs send alarms. There is no way to selectively deactivate an individual DI's alarms using session control commands. Instead, you must send the network control command CANCEL_SOURCE_ALARM_MESSAGE to the DI and specify the appropriate alarm message numbers (see command description in chapter 8). This command turns alarm messages off for all operators, because it directs the DI not to send the alarm.

Routing Command Responses and Alarms

You can route command responses and alarms to files other than your display screen using standard NOS/VE files and commands. Routing responses and alarms to files can help you keep a record of responses and alarms. You can review the files and print them, if necessary. Routing is helpful with lengthy responses, such as the responses to the display status and configuration network control commands, which may return several screens of data.

To route responses and alarms to files, use the SCL command CREATE_FILE_CONNECTION. Refer to the SCL for NOS/VE System Interface manual for a complete description of this command. CREATE_FILE_CONNECTION establishes a connection between one of the standard NOS/VE files and one or more files. Any data written to the standard file is also written to the file you specify. The allowed standard file names include the following.

$ECHO
$ERRORS
$INPUT
$LIST
$OUTPUT
$RESPONSE

Routing Responses

Normal command responses are written to the file specified on SEND_COMMAND. The default output file is the standard NOS/VE file $OUTPUT. Error responses are written to standard NOS/VE file $RESPONSE. Use the CREATE_FILE_CONNECTION command to connect a file to these standard files. If you only want a file of error messages, specify $RESPONSE.

NETOU commands and any NOS/VE commands you enter are written to standard file $ECHO. For a complete record of your operations sessions which include both commands and responses, use the CREATE_FILE_CONNECTION command to connect a file to $OUTPUT, $RESPONSE and $ECHO. You can use the standard job log file ($JOB_LOG) to serve as the file to which all commands and responses are written. The job log adds a date and time stamp to the commands and responses. By default, $RESPONSE is connected to $JOB_LOG.

Routing Alarms

All alarms are written to the file specified on ACTIVATE_ALARMS. The default output file is the standard file $OUTPUT. For an alarm history file, use the CREATE_FILE_CONNECTION command to connect another file to $OUTPUT, or to any other file you specify as the one to receive alarm output on. You can write the alarms to the same file to which responses are written.

Accessing Response and Alarm Files

Use the standard NOS/VE commands for accessing and displaying the files to which responses and alarms are written. If you write responses and alarms to $JOB_LOG, use the DISPLAY_LOG command to display the job log.

Responding to Alarms

Check the CDCNET Diagnostic Messages manual for the description of the alarm you have received and the suggested actions for each message. Alarms may also be messages to you from a terminal user. If the alarm is a message from a terminal user, send a message back to the terminal user by the same line name listed in the alarm, using the WRITE_TERMINAL_MESSAGE command. (See chapter 8, Network Commands).

Session Control on NOS

This section describes how to use commands to control your CDCNET network operations sessions in a NOS environment.

Session Control Commands

The following is a summary of the session control commands used exclusively with NOS-based CDCNET operations environments. The commands are described in chapter 6.

Table 3-1. NOS Session Control Commands

Command	Description
ACTIVATE_ALARMS (ACTA)	Initiates receipt of alarms from all DIs and DI communities at an operations station. This command is also used in NOS/VE operations environments.
CHANGE_ALARM_ ENVIRONMENT (CHAAE)	Changes the list of DIs from which an operations station receives alarms. You may shut off or turn back on the receipt of alarms from individual DIs. This command does not affect alarms received by other network operators, if your network has more than one network operator.
DEACTIVATE_ALARMS (DEAA)	Terminates receipt of alarms from all DIs at an operations station. This command is also used in NOS/VE operations environments.
DISPLAY_ALARM_ ENVIRONMENT (DISAE)	Displays the alarming status of the DI communities (enabled or disabled), and the list of DIs for which alarms are disabled.
DISPLAY_ALARM_HISTORY (DISAH)	Displays a maximum of 50 lines of alarms received at your operations station.
DISPLAY_CATENET_TITLES (DISCT)	Displays the DI community, system, and service titles in the catenet that are registered through the Directory Management Entity.
DISPLAY_COMMAND_ INFORMATION (DISCI)	Displays the parameters and parameter syntax for the specified session command.
DISPLAY_COMMAND_LIST (DISCL)	Displays an alphabetical list of the network operator commands for which you are validated.
DISPLAY_COMMAND_LIST_ ENTRY (DISCLE)	An alias for the DISPLAY_COMMAND_LIST command.
DISPLAY_CONNECTED_MDI (DISCM)	Displays the system titles and connection status of the MDIs or MTIs physically connected to the host mainframe.

(Continued)

Table 3-1. NOS Session Control Commands *(Continued)*

Command	Description
EXECUTE_COMMAND_FILE (EXECF)	Executes CDCNET network operations commands in the specified file.
HELP	Performs the same function as the DISPLAY_COMMAND_LIST command. Refer to the DISPLAY_COMMAND_LIST description.
INCLUDE_FILE (INCF)	Provides the same functions as EXECUTE_COMMAND_FILE command. Refer to the EXECUTE_COMMAND_FILE description.
QUIT	Terminates the Network Operator Utility (NETOU) session.
RESTORE_ALARM_ENVIRONMENT (RESAE)	Restores a changed alarm environment to its original definition that was defined when you first logged in to NETOU. Reenables receipt of alarms from DIs at an operations station.
ROUTE_ALARM (ROUA)	Routes all alarms to a specified file.
ROUTE_COMMAND_RESPONSE (ROUCR)	Routes all command responses to a specified file.
SEND_COMMAND (SENC)	Sends the CDCNET command to a DI or list of DIs.
SEND_COMMAND_SEQUENCE (SENCS)	Allows you to send one or more commnds to the same system(s) without enveloping the command within a SENC command.
SET_COMMAND_MDI (SETCM)	Selects the MDI or MTI through which you send commands and through which you receive responses and alarms from the network. At any time, you can communicate with only one MDI or MTI. (See Selecting an MDI or MTI in chapter 2 for use of the SETCM command.)
**	Terminates the SEND_COMMAND_SEQUENCE execution mode.

Session Control Activities

This section provides instructions for using session control commands to set up and control your operations sessions in NOS environments.

Using a Prolog

A prolog is a file containing a list of commands that are executed each time an activity is initiated. The prolog file is a NOS indirect access file, residing in your operator's catalog with the file name of NETOPRP. (Note: The file name containing your prolog cannot be changed and has to remain NETOPRP.) You can create a prolog for your NETOU sessions that will be executed every time you access NETOU. You can put any NETOU or NOS command in your prolog file. Typically, your user prolog contains the CDCNET commands to establish your command environment. Instead of entering these commands every time you access NETOU, you put the commands in your prolog to establish your command environment whenever you invoke NETOU. You could also include a SEND_COMMAND_SEQUENCE command in your prolog. This would save you typing as you would not have to enclose each command within the SENC command.

Using Command Files

Command files are files containing CDCNET network operations commands (both Session Control commands and the commands that monitor, control, and configure DIs). You can build command files to perform session and network control activities. The commands in the file are executed as if it were an individual command you typed in at your operations site. A break sequence will terminate command file processing.

Command files can be an efficient way to send commands and save keyboard entry, since several commands that perform a single activity can be grouped together in a file. Once a command file is created and saved, when you need to perform an activity for which the command file was created, you can call the command file and execute it using the EXECUTE_COMMAND_FILE command, rather than entering all the commands individually. Chapter 4 describes network operations activities and commands that perform the activities. You can build command files to perform the activities described there.

NOTE

Some commands and procedures used to perform network operations activities are not a part of NETOU, but run under another application. You may not include commands and procedures that are not CDCNET network operations commands in command files. Commands and procedures that are described in this manual but are not allowed in CDCNET network operations command files include:

- Network_Logfile_Termination Utility (NLTERM).
- Network_Logfile_List (NLLIST).
- All Network Performance Analyzer (NPA) commands and procedures used to obtain network statistics.

Writing Command Files

The following procedure assumes that you have access to an editing program, such as NOS Full Screen Editor (FSE). Command files can either be created at a host console or an interactive terminal. However, because interactive terminals with full screen interface are better suited to file editing, this procedure is geared toward an interactive terminal using FSE.

1. CDCNET command files must be written in the NOS 6/12 ASCII character set. To ensure this, enter the NOS ASCII command prior to accessing FSE.
2. Create a NOS local file under FSE.
3. Using FSE, enter the appropriate session and network commands in the file. The commands EXECUTE_COMMAND_FILE, INCLUDE_FILE and SET_COMMAND_MDI cannot be used in command files. To put comments in the command file, enclose the comment text in quotation marks.
4. Make the command file an indirect access permanent file using the SAVE command.

   SAVE,file_name

   It is recommended that you make a note of the command file's purpose, either in the file itself or in your records. This is important if you have many command files or several modified versions of a command file.

5. Test the file by attempting to execute it using the EXECUTE_COMMAND_FILE command (refer to Executing Command Files).

Executing Command Files

To execute a command file, enter the EXECUTE_COMMAND_FILE command.

EXECUTE_COMMAND_FILE FILE=file_name,USER_NAME=name

Provide the name of the command file you want to execute. USER_NAME is optional. Use it if the command file is not in your permanent file catalog, but under another user name. In that case, the file must be public or semi-private, as you must have permission to access the file.

The following command file sends a set of display status commands to a list of three DIs, MDI1, TDI1 and TDI2 (except for line status, which is only sent to the TDIs). For more information on the display status commands, see chapter 4.

```
"File STATUS displays status of DI hardware and software."
sencs s=(mdi1, tdi1, tdi2) display_di_system_status'
display_hardware_status,
display_line_status,
display_network_status,
display_software_load_status,
display_xns_transport_status,
display_directory_status,
```

The following command file, DEFETH, is a standard set of commands used to logically reconfigure an Ethernet network solution. Parameter values are left blank so the file can be copied and parameter values can be specified. A command file is useful in this situation because defining and starting a network solution involves defining and starting the network at two places. Once a file of commands to define and start a network solution is created, the file can be duplicated and can be used to define and start the network solution on each DI affected by the definition change. This command file includes comments that describe the file's use.

```
"File DEFETH"
"This file is a template file of network operations commands"
"that can be copied and used to define an Ethernet network solution."
"Insert the appropriate parameter values where indicated."
"Not all optional parameters are shown.  If other parameters are added"
"to the command being sent, they must be placed within the final"
"apostrophe character."

Send_command Command='Stop_network Network_name=      'System=

Send_command Command='Cancel_ether_net Network_name=  ',System=

Send_command Command='Define_ether_trunk Slot=  ,Trunk_name=',System=

Send_command Command='Define_ether_net,..
  Trunk_name=      ,Network_ID=       Network_name=        ',System=

Send_command Command='Start_Network Network_name=      ',System=
```

The following EXECUTE_COMMAND_FILE example executes a file called TRMSTAT, that starts collection and reporting of line statistics. The file TRMSTAT is under another user name, so an alternate user name is specified with the command.

```
EXECUTE_COMMAND_FILE FILE=TRMSTAT,UN=ZELDA
```

Routing Command Responses and Alarms

You can route command responses and alarms to a file using the ROUTE_COMMAND_RESPONSE and ROUTE_ALARM commands. Routing of responses and alarms allows you to review responses, retain them in a NOS permanent file, and print the file to more thoroughly review the responses. Routing is helpful with lengthy responses, such as status and configuration displays, which may return several pages of data.

To route responses, enter:

ROUTE_COMMAND_RESPONSE FILE = (file_name,DISPLAY) or DISPLAY or file_name

To route alarms, enter:

ROUTE_ALARM FILE = (file_name,DISPLAY) or DISPLAY or file_name

Specify a file name as the file to receive the responses or alarms. This file must be a NOS direct access permanent file. If the file does not exist when the command is executed, a new file will be defined. If the file does exist, responses or alarms will be appended to the end of the file. If you enter DISPLAY, command responses or alarms are routed to your operations station. If you enter DISPLAY without any parameters, command responses or alarms are routed to your operations terminal (DISPLAY is assumed). If you specify a file name, but do not enter DISPLAY, command responses or alarms are not routed to your operations station. At the start of your session, routing of responses to your operations station (DISPLAY) is assumed.

You may simultaneously route command responses or alarms to your display and to a file by specifying both DISPLAY and another file name as a list with the command. You may also route command responses and CDCNET alarms to the same file.

When requesting the status of several DIs and lines, you could create a file called NSTATUS to receive the status responses, and route the responses to NSTATUS by entering:

```
route_command_response file=nstatus
```

The following command example directs all alarms to a file named OPALARM and to the operations station.

```
route_alarm file=(opalarm,display)
```

Accessing Routed Responses and Alarms

To access files containing CDCNET command responses and alarms log into IAF or switch to your IAF connection by the CHANGE_WORKING_CONNECTION terminal user command, if you have established multiple connections at your operations station. Use the NOS command ATTACH to attach the file, and the Full Screen Editor to view the file. You may also route the file to a printer using the NOS command ROUTE. Refer to the NOS Reference Set, Volume 3 for the format of the ROUTE command.

Displaying Alarm Environment

The DISPLAY_ALARM_ENVIRONMENT command shows the current alarm reporting set-up for your operations station. Refer to the DISPLAY_ALARM_ENVIRONMENT command description in chapter 7 for the display the command generates.

Changing Alarm Environment

To change the alarm reporting set-up for an operations station, enter the CHANGE_ALARM_ENVIRONMENT (CHAAE) command. This command will change the list of DIs that send alarms to you. The CHANGE_ALARM_ENVIRONMENT command also enables alarms.

To shut off alarms from a DI, enter:

CHANGE_ALARM_ENVIRONMENT DISABLE_SYSTEM= DI name or names

To turn alarms from a DI back on, enter:

CHANGE_ALARM_ENVIRONMENT ENABLE_SYSTEM=DI name or names

NOTE

The CHANGE_ALARM_ENVIRONMENT command is effective only for the operator who enters the command. If there is more than one network operations station active at your site, the alarms will still go to the other operators. If you want to turn off alarms for all operators, cancel the source alarm messages at the individual DIs using the CANCEL_SOURCE_ALARM_MESSAGE command.

There are two other commands that can be used to activate and deactivate receipt of all alarms from *all* DIs at an operations station: ACTIVATE_ALARMS and DEACTIVATE_ALARMS. You cannot selectively enable or disable alarms with these two commands; use CHANGE_ALARM_ENVIRONMENT and specify the DIs for which you want to activate alarms.

The ACTIVATE_ALARMS command activates receipt of alarms from *all* DIs in the catenet at an operations station. The effect of ACTIVATE_ALARMS is the same as using the CHANGE_ALARM_ENVIRONMENT command to enable all alarms in the CATENET community of DIs. On NOS, alarms are activated by default. You do not need to use an ACTIVATE_ALARMS command to enable alarm reporting at your operations station at the beginning of your NETOU session.

The DEACTIVATE_ALARMS command deactivates receipt of alarms from *all* DIs in the catenet. The effect of DEACTIVATE_ALARMS is equivalent to using CHANGE_ALARM_ENVIRONMENT to disable all alarms in the CATENET community of DIs.

Restoring Alarm Environment

Use the CHANGE_ALARM_ENVIRONMENT command to add DIs back to the list of DIs that report alarms to you, or use the RESTORE_ALARM_ENVIRONMENT command. The RESAE command restores all DIs to the list of DIs reporting alarms to you. This list of DIs was originally defined at the beginning of your operations session.

Displaying Alarm History

The DISPLAY_ALARM_HISTORY command displays the alarms received at your operations station since the start of your NETOU session.

DISPLAY_ALARM_HISTORY DISPLAY_OPTION=option

The options for this command are LAST, PAGE, and ALL. LAST displays all alarms received since the last DISAH command was entered. PAGE displays the last page of alarms received. ALL displays all alarms received in the alarm history buffer, which is limited by buffer size to 50 lines of display. If the buffer receives more than 50 lines of display, new lines of display are written over the oldest alarms in the file. Because there is a blank line between each alarm, you may see only 34 non-blank lines of text.

For example,

```
display_alarm_history
```

returns this display.

```
ALARM HISTORY REPORT

****** ALARM FROM MTI_83              85/10/10  13.38.51     619
--ERROR--   Line: LINE31 down, connection timer expired

****** ALARM FROM MTI_83              85/10/10  13.38.55     202
--ERROR--   Line: LINE23 down, auto-recognition failed

****** ALARM FROM MTI_83              85/10/10  13.40.28    202
--ERROR--   Line: LINE23 down, auto-recognition failed
```

Responding to Alarms

Check the CDCNET Diagnostic Messages manual for the description of the alarm you have received and the suggested actions for each message. Alarms may also be messages to you from a terminal user. If the alarm is a message from a terminal user, send a message back to the terminal user by the same line name listed in the alarm using the WRITE_TERMINAL_MESSAGE command.

Network Control 4

Network Control 4

This chapter contains instructions for performing CDCNET network control activities. Network control activities use network commands to monitor, control, and dynamically reconfigure network equipment. Activities are divided into basic and advanced categories.

The following basic activities are likely to be performed on a regular basis by network operators and do not require extensive knowledge of network configuration and software:

- Recordkeeping: Keeping a database of all the network equipment (DIs, lines, trunks) and their locations.
- Checking status of network components.
- Starting and stopping communications on communications lines.
- Starting and stopping communications on network solutions.
- Sending messages to terminal users.
- Receiving messages from terminal users.
- Network clock management: Synchronizing DI time clocks.
- Running CDCNET online diagnostics.
- Displaying logical configuration of network components.

The following advanced activities require a deeper understanding of the network, its configuration, and software that runs in DIs and on host computers. Advanced activities include procedures that may affect the performance of the network, such as canceling the logical configuration of a communication line or resetting a DI. Such activities are usually performed by an analyst or by an operator under an analyst's supervision.

- Changing the network's logical configuration.
  - Adding or deleting a communication line.
  - Adding or deleting a network solution.
  - Adding terminal devices.
  - Adding batch devices.
  - Starting and stopping a gateway.
  - Logging and alarm control.
    - Defining log messages to be generated by a DI.
    - Defining alarm messages to be generated by a DI.
    - Terminating and archiving CDCNET network log files.

- Statistics control.

  Starting and stopping statistics collection and reporting.
  Obtaining statistics results.

- Running NPA procedures (refer to NPA manual).

- Resetting and dumping a DI.

- Loading and unloading software.

For many of these activities, you can use command files to simplify command entry. Refer to chapter 3 for more information on command files.

Basic Operations Activities

Basic activities are network control activities you will most likely perform on a regular basis.

Recordkeeping

Keeping track of the network's components, their locations, and their maintenance schedule is an important part of network operations.

Your recordkeeping system should include:

- A diagram of the network's physical layout. The diagram should note the location of all equipment at your site (mainframes, DIs, Ethernet cables, communication lines, hardwired (dedicated) terminals, dial-up (switched) lines, and other network equipment).

- A current list of the logical names assigned to network components. The names for physical components (DIs, network solutions, communication lines) should be shown on the network diagram. When configuration changes or replacements are made, be sure to update this list. You can use the following commands to generate lists of the current logical names and titles defined for the network: DISPLAY_LOGICAL_NAMES, the $MATCHING_NAMES function on NOS/VE, and DISPLAY_CATENET_TITLES on NOS.

- The channel number and mainframe ID for the mainframe connected to an MDI or MTI.

- A list of the serial numbers assigned to DIs. These should also be included on the network diagram.

- Dial-up connections and their baud rates.

- A list of all ports for each DI and each line connected to the DI.

- Maintenance records for all DIs including diagnostic results, repairs, and replacements; problems reported to operator, and records of customer engineer visits.
- The DI, for DIs supported by NOS hosts, that contains the catenet's master clock (from which all other DIs set their clocks). The location of the master clock is determined during configuration when the functions for each DI are defined in the DI's configuration file. A DI that contains the master clock is known as a clocking system. For DIs supported by NOS/VE systems, the master clock is configured in a NOS/VE host.
- NPA reports.

NOTE

If you do not have a record of which DI contains the master clock, send a DISPLAY_SYSTEM_OPTIONS (DISSO) command to each DI in the network. Specify with the DISPLAY_OPTION (DO) parameter that you want the DI to return a display of whether or not it is a clocking system.

```
SEND_COMMAND COMMAND='DISSO DO=CLOCKING_SYSTEM',SYSTEM=di_name
```

The DI that returns

```
Clocking system=yes.
```

contains the master clock for the network. Mark the location of the master clock in your records and on the network diagram.

You may find it helpful to attach tags to your DIs listing:

- Mainframe (where applicable, as in MDIs and MTIs).
- Mainframe channel number (where applicable).
- Ethernet trunk (where applicable, as in MDIs and TDIs).
- DI type (MDI, MTI, TDI, NDI, RTI).
- DI serial number.
- DI system ID.

Such information will be helpful for people at your site who are unfamiliar with CDCNET hardware, and for you when dealing with CDCNET network problems over the phone.

You can develop an online recordkeeping database of information about network components such as DIs, circuits, lines, ports, locations and logical names, by using the configuration files for the DIs. Include the previously listed information as comments in the configuration files for your DIs. You can also include comments such as the system ID, the DI's location, and the original date of installation and of subsequent configuration changes. Print copies of the configuration files regularly and arrange them in a binder.

It is important to update the map and the database regularly, particularly when configuration changes, problems and repairs occur. If your site has several network operators, be sure to keep each other informed about changes to the records.

Checking Status of Network Components

In this activity, you request and obtain the current operational status of network components, such as hardware boards in a DI, communication lines, network solutions, and DI software, using the display status commands. The status is returned to you as a display. You may route the displays to a file using SCL commands on NOS/VE or by the ROUTE_COMMAND_RESPONSE session control command on NOS (see chapter 3).

A status display is similar to a snapshot in that it gives a picture of how the network is running at the time that the status command is processed. While NPA reports give more thorough and extensive reports than status displays, they are run at intervals and show the performance of the network over time. Status displays can be requested and received anytime the network is running, and show how the network component is performing at the time you request the status. This "snapshot effect" is important when you are investigating user-complaints or problems with the network. In such situations, you need to isolate the problem and return the user to network services as quickly as possible. Checking network component status is a first step in this process.

Check the status of network components by entering display status commands. Display status commands display the operational status of the hardware devices, communication lines, network solutions, and communication software configured for a DI system. For a complete description of these commands, see their descriptions in chapter 8.

Command	Description
DISPLAY_DEVICE_OUTCALL_STATUS	Displays the current status of the Device Outcall Service.
DISPLAY_DI_SYSTEM_STATUS (DISDSS)	Returns general information about the DI's operating system and its memory and buffer usage. Information includes date and time of the last reload, version of load file used, states of buffers and memory, and CPU usage.
DISPLAY_DIRECTORY_STATUS (DISDS)	Displays the operating status of the Directory Management Entity in a DI.
DISPLAY_HARDWARE_STATUS (DISHS)	Displays status of boards, ports, and memory banks in a DI.
DISPLAY_LINE_STATUS (DISLS)	Displays status of terminal communication lines and the connections established for these lines.
DISPLAY_NETWORK_STATUS (DISNS)	Displays status of network solutions to which a DI is connected.
DISPLAY_PASSTHROUGH_STATUS (DISPS)	Displays the configuration of the interactive passthrough gateway and the status of all connections supported by the gateway.
DISPLAY_ROUTING_STATUS (DISRS)	Displays the operating status of the Routing Management Entity in a DI.

Command	Description
DISPLAY_SOFTWARE_LOAD_STATUS (DISSLS)	Displays whether or not software modules are loaded in a DI.
DISPLAY_XNS_TRANSPORT_STATUS (DISXTS)	Displays the operating status of the XNS (Xerox Networking Software) Transport layer software, the status of the specific service access points (SAPs) serviced by XNS Transport and the status of specific connections serviced by XNS Transport.

Starting and Stopping Communication Lines

These activities involve controlling the communications traffic on each specific communication line. Before performing these activities, make sure you know the network's physical configuration and the logical names assigned to the network's communication lines.

Starting and stopping lines may be done for several reasons, such as replacing a communication line and changing a line's logical configuration. Stopping a communication line will cut off a terminal user from the rest of the network. If you have to stop a line connected to a terminal, inform the terminal user well in advance that the line will be stopped by sending a WRITE_TERMINAL_MESSAGE command to the terminal user (see description in chapter 8).

Starting a Line

To start an individual line, you must know the line and the logical names of the DI supporting the line. You may use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines (see command description in chapter 8).

Requirements:

- The line must be defined in the network's configuration by the DEFINE_LINE (DEFL) command. (A configured line is a line that has been assigned to a specific terminal interface program (TIP) that will service the line when the line is started. If the line is not configured, a TIP has not been assigned to start and service the line.) If you're not sure the line is configured, check the DI's configuration file or enter the DISPLAY_LINE_STATUS command. Configured lines will be indicated by `configured` in the command response.

- The terminal interface program (TIP) supporting this line must be configured by the DEFINE_TIP (DEFT) command. Check the DI's configuration file for this command.

  To start a line, enter the START_LINE command.

  START_LINE LINE_NAME=line_name

Example:

```
send_command command='start_line line_name=group_1',system=first_floor_tdi
```

Stopping a Line

To stop an individual line, you must know the logical name of the line. You can use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines (see command description in chapter 8). Use the following procedure.

1. Notify the line's user that the line will be stopped using the WRITE_TERMINAL_MESSAGE command (see command description in chapter 8). Tell user to log off.

2. Enter the STOP_LINE command.

   STOP_LINE LINE_NAME=line_name

Example:

```
send_command command='stop_line line_name=line23',system=main_tdi
```

Starting and Stopping Network Solutions

These activities affect a larger part of the CDCNET network than starting and stopping communication lines. Stopping a network solution logically removes a portion of the CDCNET network over which data can travel.

Network operations commands exist to start and stop communications over Ethernet, X.25, HDLC, and channel network solutions.

Do not stop the network solution that connects the operations station to the network host computer. Stopping the network solution which connects to the TDI that supports the operations station leaves the TDI (and you) logically disconnected from the network.

For example, if a TDI is connected to a CDCNET over a single Ethernet network solution, you should not stop communications on that network solution, because it is required to carry operations commands and other data to the TDI. You will not be able to start the network solution again unless you manually reset the TDI.

Starting a Network Solution

Starting the network solution also starts the underlying trunk, if not already started.

Requirements:

- The network solution must be defined by the appropriate network definition commands. See Adding and Deleting Network Solutions in the Network's Logical Configuration in the following section, Advanced Operations Activities.

- Know the network solution's logical name as it is defined for the DI to which you will be sending the commands. Use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines (see command description in chapter 8).

Enter the START_NETWORK command.

   START_NETWORK NETWORK_NAME=network_name

Example:

```
send_command command='start_network network_name=net_1',system=tdi04
```

Stopping a Network Solution

Requirements:

- Check the network's physical and logical configuration to determine the connections between DIs and network solutions. Do not stop a network solution if it is the only network solution over which your commands can be sent to a DI.

- Know the network solution's logical name.You may use the DISPLAY_LOGICAL_NAMES command to determine the logical names for DIs and lines (see command description in chapter 8).

Enter the STOP_NETWORK command.

STOP_NETWORK NETWORK_NAME=network_name

The STOP_NETWORK command stops the underlying trunk if the network solution is the only traffic being carried by the trunk.

Example:

```
send_command command='stop_network network_name=net_1',system=tdi04
```

Sending Messages to Terminal Users

You can send messages to terminal users by sending the WRITE_TERMINAL_MESSAGE command through NETOU.

This command allows you to send messages to all users, to users connected to a particular service, or to a particular line. You enclose the message within quotation marks. The optional parameters LINE_NAME, DEVICE_NAME, and SERVICE_NAME allow you to specify where you want the message to go. You may send a message to a specific line or group of lines, to a particular terminal device or group of devices, or to the users of a specific gateway service. For example, if you send a message specifying a particular NOS/VE or NOS service name with the SERVICE_NAME parameter, all terminal users currently connected to the service name specified will receive the message. Only terminals that match the parameters you specify will receive this message. If you do not specify the optional parameters, the message is sent to all terminal users.

The message you specify with the message parameter must be entered as a string value enclosed by two consecutive apostrophes. If you want the message to have several lines of text, you must enter each line to be output at the terminal as a string value within parentheses, as in the following example.

The following command sends a message to a terminal user connected to TDI1 and on a line called LINE15:

Example:

```
send_command c='write_terminal_message,..
message=(''New communications configuration tomorrow'',''Network down ..
until 10:00.''),line_name=line15',system=tdi1
```

Receiving Messages from Terminal Users

Messages from terminal users are sent to the network operator by a terminal user command called REQUEST_NETWORK_OPERATOR (REQNO). These messages show up at your operations station as alarms. On NOS, a warning bell will ring at an interactive terminal, and NETOU will be displayed on the operator attention line at the host console.

The alarm message from a terminal user gives the line name, terminal device name, gateway service through which the message was sent, and the text of the message. You can route terminal user messages to a file using standard SCL commands on NOS/VE or the ROUTE_ALARM command on NOS (see chapter 3).

It is recommended that you send a message back to the user using the WRITE_TERMINAL_MESSAGE command to acknowledge that you have received the message.

Example:

```
****** ALARM FROM  riverside_tdi_1                   85/06/13 11.15.45 168
Terminal User Request
line_name = mech_eng_2
Device name = mech_eng_term_2
Message: Will be moving office next week.  Need configuration change form.
```

NOTE

The REQNO command does not execute successfully (a terminal user cannot contact the network operator using this command) unless CDCNET log message number 168 is enabled as an alarm by the DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM) configuration command on NOS or by ACTIVATE_ALARMS command (ACTA) ON NOS/VE. Message number 168 is enabled as an alarm by default. Refer to Logging and Alarm Control in this chapter.

Network Clock Management: Synchronizing DI Time Clocks

Each DI has a clock that maintains the date and time for the DI. The date and time set at a DI are added to log messages and alarms generated by a DI. So that log messages and alarms from different DIs can be correlated, clock management functions ensure that all DIs in a catenet are synchronized (within one second of each other). There are two parts to the clock management function: Resetting the master clock for the catenet and synchronizing all of the clocks in the catenet to the date and time set at the master clock. For CDCNET networks supported by a NOS/VE host, the master clock is configured in the NOS/VE host. For CDCNET networks supported by a NOS host, the master clock is configured in a DI in the network. This DI is called the clocking_system DI.

Network Clock Management involves the following activities.

- Resetting the master clock, using the SET_DATE_AND_TIME command (NOS, only).
- Synchronizing time clocks in all DIs, using the SYNCHRONIZE_CLOCK command.
- Displaying date and time set at a DI, using the DISPLAY_DATE_AND_TIME command.

Resetting the Master Clock (NOS Only)

1. Determine which DI contains the master clock by one of the following methods.

   - Check your site's records and network map (if available) for the DI marked as containing the master clock.

   - Send a DISPLAY_SYSTEM_OPTIONS (DISSO) command to each DI, specifying CLOCKING_SYSTEM with the DISPLAY_OPTION parameter. Enter:

     ```
     senc c='disso do=clocking_system',system=mdi_1
     ```

   The DI that contains the master clock sends the following response.

     ```
     clocking system = yes
     ```

2. Once you have located the DI containing the master clock, reset the master clock by sending a SET_DATE_AND_TIME command to that DI. Provide the current date and time for the DATE and TIME parameters. Both the date and time must be entered as string values enclosed by two consecutive apostrophes, as in the following example. Refer to the SET_DATE_AND_TIME command description in chapter 8.

   Example:

   The master clock for a network is located in a DI called TDI2. To reset the master clock, the operator sends a SET_DATE_AND_TIME command to TDI2.

   ```
   senc c='setdat d=''24/11/85'',t=''08:25:49''',s=tdi2
   ```

After the master clock has been reset, synchronize all the DI clocks using the SYNCHRONIZE_CLOCK command, as described in this section.

Synchronizing Time Clocks in All DIs

The clock synchronization automatically occurs when a DI is configured. Once a day, all DI clocks should be resynchronized. Over one day's time, for example, clocks could be running one to two seconds out of synchronization with each other. The SYNCHRONIZE_CLOCK (SYNC) command synchronizes a DI's clock to the master date and time set at the master clock.

To synchronize the DI clocks, send the SYNCHRONIZE_CLOCK command to every DI in the network or write and execute a command file that sends SYNCHRONIZE_CLOCK to every DI in the network (refer to chapter 3 for directions on writing a command file).

Displaying Date and Time Set at a DI

If, at any time, you want to see the date and time set at a DI, send the DI a DISPLAY_DATE_AND_TIME command.

Example:

```
senc c='display_date_and_time',s=north_tdi_1

System date and time
    24/11/86 08:25:49
```

Running CDCNET Online Diagnostics

Diagnostic control commands place physical devices under diagnostic control and start or stop online diagnostics on these devices. To use online diagnostics, send online diagnostics control commands to the DIs containing the devices you want to test.

The online diagnostics test the following hardware: Communications Interface Module (CIM), Line Interface Module (LIM), LIM ports, Ethernet Serial Channel Interface (ESCI), Mainframe Channel Interface (MCI), and Unit Record Interface (URI). When testing the above hardware devices, you must also stop communications traffic for the board or port.

An online diagnostics test affects only the board being tested. Operations and communications traffic for other boards or ports are unaffected. However, during a test, the board or port is not available for normal communications traffic. This means that you may not execute online diagnostics on the only board or port supporting the network solution over which the DI receives operations commands from you. This restriction is enforced through the STOP_NETWORK command, since communications must be stopped on the device being tested before the diagnostics can be executed.

If errors are detected during an online diagnostics test, refer to the chapter on isolating failures in the CDCNET Troubleshooting Guide.

NOTE

If you are using NOS/VE, you can use the Concurrent Maintenance Library for the Virtual Environment (CML/VE) to run the online diagnostics. CML/VE provides you with a set of menus from which you select the appropriate tasks needed to run the dianostics. For information on how to use CML/VE, refer to the CML/VE reference manual listed in the Additional Related Manuals.

Online Diagnostics Procedure

Use the following sequence to run online diagnostics.

1. Notify users that you will be stopping traffic on the line or network solution connected to the device being tested (see Sending Messages to Terminal Users in this chapter).

2. Use one of the stop communications commands to stop the communications traffic for the device to be tested. The following table shows the various stop commands that are used for each board.

Board	Command	Comments
CIM	STOP_LINE	
	STOP_NETWORK	Used if line was defined as a HDLC network solution.
	STOP_X25_INTERFACE	Used if line was defined as an X.25 network solution.

Board	Command	Comments
LIM	STOP_LINE	
	STOP_NETWORK	Used if line was defined as a HDLC network solution.
	STOP_X25_INTERFACE	Used if line was defined as an X.25 network solution.
ESCI	STOP_LINE	
MCI	STOP_NETWORK	

3. Place the device to be tested in a down state by sending the CHANGE_ELEMENT_STATE command to the device.

4. Start the diagnostic test using the appropriate start diagnostic command. Send the diagnostic command within a SEND_COMMAND to the DI which you are testing. Start diagnostic commands include:

   START_CIM_TEST

   START_ESCI_TEST

   START_LIM_TEST

   START_MCI_TEST

   START_PORT_TEST

   START_URI_TEST

   For command descriptions and parameters, see the individual command descriptions in chapter 8.

5. Monitor the online diagnostics test results by sending the DISPLAY_TEST_STATUS command to the DI containing the device being tested. You can also monitor the test results by enabling the log messages defined for online diagnostics as alarms. The message numbers currently defined and used for reporting online diagnostics events are 337..352. Enter the DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM) command and specify the message numbers to be enabled. To disable the alarm messages, enter the CANCEL_SOURCE_ALARM_MESSAGE (CANSAM) command. Refer to chapter 8, Network Commands, for command descriptions.

6. To terminate the diagnostics before all the passes of the test are completed, enter the appropriate stop diagnostic command. You may not have to use a stop test command unless you are running many passes of the test. Stop diagnostic commands include:

   STOP_CIM_TEST

   STOP_ESCI_TEST

   STOP_LIM_TEST

   STOP_MCI_TEST

STOP_PORT_TEST

STOP_URI_TEST

See chapter 8 for command format.

7. When the diagnostic completes and you have fixed the problem or no error was detected, enter a start communications command (START_NETWORK and START_LINE) to restart the communications traffic to the device being tested.

For more information on CDCNET online diagnostics and how they can be used to troubleshoot DIs, refer to the CDCNET Troubleshooting Guide.

Running CDCNET Inline Diagnostics

The inline diagnostics tests the Mainframe Channel Interface (MCI). The inline diagnostics test must execute while normal communications traffic continues. To use the inline diagnostics, send the inline diagnostics control command to the DI containing the device you want to test.

Inline Diagnostics Procedure

Use the following sequence to run inline diagnostics.

1. Notify users that you will be executing an inline diagnostics test on the MCI to which their line or network solution is connected (see Sending Messages to Terminal Users in this chapter).

2. Start the diagnostic test using the START_MCI_INLINE_TEST diagnostic command. Send the diagnostic command within a SEND_COMMAND to the MCI which you are testing.

3. To terminate the diagnostic before the test completes, enter the STOP_MCI_INLINE_TEST diagnostic command.

4. The diagnostic executes once, stopping after the first pass.

For command descriptions and parameters, see the individual command descriptions in chapter 8. For more information on the MCI inline test, refer to the CDCNET Troubleshooting Guide.

Displaying Configuration of Network Components

Each DI is logically configured through network configuration commands, which are in configuration files residing in the CYBER host. The display configuration commands display the current values of parameters on network configuration commands. Using this display, you can observe current configurations and decide if the existing configurations should be changed.

The following are display configuration commands.

Command	Description
DISPLAY_CHANNEL_NET_OPTIONS	Displays MCI channel network attributes.
DISPLAY_CHANNEL_TRUNK_OPTIONS	Displays the configuration of an MCI trunk.
DISPLAY_DEVICE_OUTCALL_STATUS	Displays the current status of the Device Outcall Service.
DISPLAY_ETHER_NET_OPTIONS	Displays the Ethernet network attributes.
DISPLAY_FILE_SUPPORT	Displays the file types for which file access is supported through the destination system or systems for the command. Used only for MDIs or MTIs connected to a NOS host.
DISPLAY_HDLC_NET_OPTIONS	Displays HDLC network attributes.
DISPLAY_HDLC_TRUNK_OPTIONS	Displays the configuration of an HDLC trunk.
DISPLAY_LINE_OPTIONS	Displays a list of the attributes of a communications line or a URI line.
DISPLAY_NP_GW_OUTCALL_OPTIONS	Displays the configuration of an application-to-application gateway outcall to a NOS host. Displays the title or titles registered in the catenet and the associated Network Products application names.
DISPLAY_OPERATOR_SUPPORT	Displays the user names of the operators currently logged in to the Network Operator Utility (NETOU). Used only for MDIs or MTIs connected to a NOS host.
DISPLAY_RECORDER_LOG_GROUP	Displays the log groups supported by a DI acting as a recorder of log messages, and the priority of the log recording function for each log group. Used only for MDIs or MTIs connected to a NOS host.

Command	Description
DISPLAY_REMOTE_LOAD_SUPPORT	Displays the current configuration of a DI's remote load support and the current load status of the remote DI.
DISPLAY_SERVICE_DISPLAY	Displays the list of interactive service names that are included in the terminal user display_services command displayable list.
DISPLAY_SOURCE_ALARMS	Displays the list of alarms to be sent to your network operations station from a DI.
DISPLAY_SOURCE_LOG_GROUP	Displays the log groups to which the DI, acting as a source of log messages, belongs and the messages to be logged for each group.
DISPLAY_SYSTEM_OPTIONS	Displays the current value of a DI's operating system program attributes such as size and threshold values for system buffers, percentage of system main memory (SMM) allocated to buffers, default activity stack size, and amount of free memory space to be reserved for the executive program.
DISPLAY_X25_GW_OUTCALL_OPTIONS	Displays the X.25 transparent gateway outcall definition.
DISPLAY_X25_INTERFACE_OPTIONS	Displays the attributes of the X.25 interface.
DISPLAY_X25_NET_OPTIONS	Displays X.25 network attributes.
DISPLAY_X25_TRUNK_OPTIONS	Displays the configuration of an X.25 trunk.

Advanced Operations Activities

This section provides instructions for network operations activities that may not be performed on a regular basis and/or that require a more extensive understanding of CDCNET than the basic activities.

Changing the Network's Logical Configuration

The activities in this subsection alter the logical configuration of the network using network operations commands to logically add, delete, and redefine communication lines, network solutions, gateways, log messages and alarm messages.

There are several types of configuration changes. Some changes, such as the addition of new DIs and network solutions can affect the entire network and its physical appearance. Other configuration changes are less visible, but are still physical changes, such as adding more lines to a DI. Logical configuration changes are changes in the network's software, such as removing a network solution's definition from a DI's logical configuration, or changing the line speed and other attributes for a communication line. These changes are not as visible, but are no less important in affecting how the network operates. Deleting an element from a network's logical configuration is as major a change as physically removing the element. A logically cancelled element can no longer be used to send, receive, or relay data. As a network operator, you may be called upon to change a DI's logical configuration. There are two ways to change a DI's logical configuration.

1. Entering configuration commands through NETOU while the network is running.

   The same commands that are in a DI's configuration procedure (except the DEFINE_SYSTEM command) may be entered during operations. These commands change the logical configuration of the DI to which you send the command. Configuration commands are described in chapter 8. This section assumes you are making configuration changes while the network is running.

   This type of configuration change made by entering commands through NETOU is not permanent. The configuration change at a DI stays in effect until that DI is reloaded. The configuration procedures on the host remain unchanged. At reload, the original configuration procedures are loaded. If you want to make permanent changes to a DI's logical configuration, you must access the DI's configuration procedures and make the changes to the procedures. You can use the MANAGE_CDCNET_CONFIGURATION (MANCC) Utility to edit configuration procedures.

2. Changing the configuration by changing the configuration procedures.

   This type of change is more permanent because it stays in effect, even if DIs are reloaded. However, these changes will be permanent only if the system is reloaded. Refer to the CDCNET Configuration and Site Administration Guide for information on MANCC.

Additional information on more advanced configuration changes such as changing terminal configuration parameters and reconfiguring a DI's base system software can be found in the CDCNET Configuration and Site Administration Guide.

Adding or Deleting a Communication Line

When a communication line is added to the network, it must be logically defined in addition to being physically installed. This definition consists of the line's logical name and characteristics of the line (see the DEFINE_LINE command description, chapter 8).

When communication lines are removed from the network, their definition must also be removed from the network's logical definition. To do this, enter a CANCEL_LINE command.

Adding a Line

1. If a terminal interface program (TIP) has not been defined for the TDI or MTI supporting this line, define the TIP by the DEFINE_TIP command (see command description, chapter 8).
2. Define the line's configuration using the DEFINE_LINE command.

   DEFINE_LINE LIM=lim_number,PORT=port_number,TIP_NAME=name

   Provide the values for the parameters. Only the required parameters are listed above. Refer to the command description in chapter 8 for the optional parameters.
3. The line should start after the DEFINE_LINE command completes, unless the optional START parameter was set to NO. If the line does not start communications, start the line (see Starting and Stopping Communication Lines in this chapter).

Example:

```
send_command command='define_tip tip_type=asynctip',system=south_tdi_2

send_command command='define_line lim=1,port=0,..
tip_name=async,line_name=l10',system=south_tdi_2
```

Deleting a Line

1. Notify user or users that the line or lines will be stopped using the WRITE_TERMINAL_MESSAGE command.
2. Stop communications traffic on the line using the STOP_LINE command.
3. Cancel the line's logical definition using the CANCEL_LINE command.

Example:

```
senc c='stop_line line_name=engin_line_31',s=engin_tdi

senc c='cancel_line line_name=engin_line_31',s=engin_tdi
```

Redefining a Communication Line

To redefine a communication line, first cancel its current logical definition. Once the definition is cancelled, you can redefine the line using the DEFINE_LINE command. If the DEFINE_LINE command included the START=NO parameter, you must use the START_LINE command. Otherwise, the START_LINE command is unnecessary.

Enter the commands to redefine a line in the following sequence.

STOP_LINE
CANCEL_LINE
DEFINE_LINE
START_LINE

Adding or Deleting a Network Solution

When network solutions are added to the network, they must be logically defined by configuration commands for the DIs using the network solutions. The configuration commands may be entered during operations, but changes will remain in effect only until the DI is reloaded. To make permanent changes, the commands must be changed in the DI's configuration procedure (see the CDCNET Configuration and Site Administration Guide).

Adding a Network Solution or NP Interface

Adding a network solution to a network's logical configuration involves defining the trunk which will support the network solution, then defining the network solution. The commands used for this depend on what type of network solution you are defining.

- Ethernet Network Solutions

  For DIs loaded across an Ethernet medium (such as a TDI), the commands used to define an Ethernet trunk and network solution, DEFINE_ETHER_TRUNK and DEFINE_ETHER_NET, are performed implicitly by each DI's load process, and default names are assigned to the trunk and network solution. Once a DI is loaded and configured, you do not have to enter these commands through NETOU to define the Ethernet trunk and network solution. A DEFINE_ETHER_TRUNK or DEFINE_ETHER_NET command sent to such a DI fails if the trunk or network is already defined.

  1. Enter the DEFINE_ETHER_TRUNK command.

     DEFINE_ETHER_TRUNK SLOT=slot_number,..
     TRUNK_NAME=trunk_name

     Provide the number of the slot in the DI which houses the ESCI board that will support the Ethernet trunk. If the DI has only one ESCI board, the slot number for the Ether trunk is optional. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the SLOT parameter, as in $ESCI4 (ESCI board in board slot 4).

  2. Enter the DEFINE_ETHER_NET command.

     DEFINE_ETHER_NET NETWORK_NAME=network_name,..
     TRUNK_NAME=trunk_name,NETWORK_ID=integer

     Provide the logical names of the network solution and trunk, and the ID number assigned to the network solution. The trunk name must be the same as the trunk name specified in the DEFINE_ETHER_TRUNK command for the trunk to be used as a network solution.

  3. Enter the START_NETWORK command.

     START_NETWORK NETWORK_NAME=<name>

     Provide the logical name of the network assigned to the network by a define command.

  Example:

```
senc c='define_ether_trunk trunk_name=ether1,slot=4',s=mdi_2
senc c='define_ether_net network_name=ARHNET,trunk_name=ether1,..
network_id=0afbb1(16)',s=mdi_2
```

Commands:

DEFINE_ETHER_TRUNK
DEFINE_ETHER_NET
START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_ETHER_NET command.

- Channel Network Solutions

1. Enter the DEFINE_CHANNEL_TRUNK command.

   DEFINE_CHANNEL_TRUNK SLOT=slot_number,..
   TRUNK_NAME=trunk_name

   Provide the number of the slot in the DI which houses the MCI board that will support the MCI trunk. The SLOT_NUMBER parameter is optional if the DI has only one MCI board. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the SLOT_NUMBER parameter, as in $MCI2 (MCI board in slot 2).

2. Enter the DEFINE_CHANNEL_NET command (NOS/VE only).

   DEFINE_CHANNEL_NET TRUNK_NAME=name,..
   NETWORK_ID=integer

   Provide the logical names of the network solution and trunk, and the network ID number assigned to the network solution. The trunk name must be the same as the trunk name specified in the DEFINE_CHANNEL_TRUNK command for the trunk to be used as a network solution. The network ID is the CDCNET network identifier of the channel network solution. The network identifier must match the value specified in the DEFINE_NETWORK command.

3. Enter the START_NETWORK command.

   START_NETWORK NETWORK_NAME=<name>

   Provide the logical name of the network assigned to the network by a define command.

Example:

```
senc c='define_channel_trunk trunk_name=mci1,slot=7',s=mdi_2
senc c='define_channel_net network_name=ARHNET,trunk_name=mci1,..
network_id=0afbb1(16)',s=mdi_2
senc c='start_network network_name=ARHNET',s=mdi2
```

Commands:

DEFINE_CHANNEL_TRUNK
DEFINE_CHANNEL_NET
START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_CHANNEL_NET command.

- HDLC Network Solutions

1. Enter the DEFINE_HDLC_TRUNK command.

   DEFINE_HDLC_TRUNK
   LIM=lim_number,PORT=port_number,LOCAL_
   ADDRESS=integer,REMOTE_ADDRESS=integer,..
   TRUNK_NAME=name

   Provide the numbers of the LIM and port to which the HDLC line is connected and which will support the HDLC trunk. Provide the address of the local HDLC station and the address of the remote HDLC station. Both addresses are specified in digits from 0 through 9. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk_name, a default trunk_name is created from the LIM and PORT parameters, as in $LIM1_PORT3.

2. Enter the DEFINE_HDLC_NET command.

   DEFINE_HDLC_NET TRUNK_NAME=name NETWORK_ID=integer

   Provide the trunk_name, which must be the same as that specified on the DEFINE_HDLC_TRUNK command. Provide the network ID, which is the CDCNET network identifier of the HDLC network solution.

Example:

```
senc c='define_hdlc_trunk lim=1 port=1 local_address=3075551212 ..
remote_address=5006221313 trunk_name=TYMN1' s=ndi_1
senc c='define_hdlc_network trunk_name=TYMN1 ..
 network_id=1234'..s=ndi_1
```

Commands:

DEFINE_HDLC_TRUNK
DEFINE_HDLC_NET
START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_HDLC_NET command.

- X.25 Network Solutions

1. Enter the DEFINE_X25_TRUNK command.

   DEFINE_X25_TRUNK LIM=lim_number,PORT=port_number,..
   TRUNK_NAME=name

   Provide the numbers of the LIM and port to which the X.25 line is connected, and which will support the X.25 trunk. The TRUNK_NAME parameter is optional and specifies a logical name for the trunk being defined. If you do not specify a trunk name, a default trunk name is created from the LIM and PORT parameters, as in $LIM3_PORT1.

2. Enter the DEFINE_X25_INTERFACE command.

   DEFINE_X25_INTERFACE TRUNK_NAME=name,..
   PUBLIC_DATA_NETWORK=name or keyword value..
   INONLY_RANGE=range 1..4095 or
   TWOWAY_RANGE=range 1..4095 or
   OUTONLY_RANGE=range 1..4095

   The trunk name must be the same as that specified on the DEFINE_X25_TRUNK command. The INONLY_RANGE, TWOWAY_RANGE, and OUTONLY_RANGE parameters specify ranges of channel numbers allotted for incoming calls and outgoing calls. At least one of these parameters must be specified. If you specify more than one range, the ranges must be ascending in the order listed above, with no overlapping value ranges.

3. Enter the DEFINE_X25_NET command.

   DEFINE_X25_NET TRUNK_NAME=name,..
   REMOTE_DTE_ADDRESS=1..15 of string,NETWORK_ID=0..7FFFFFFF(16)

   The trunk name is the name of the X.25 trunk that will support the network solution. The remote DTE address is the hexadecimal remote data terminating equipment address for this X.25 network solution. This is typically a telephone number for the other end of the network, assigned by the network provider (such as Telenet or Tymnet) when a site subscribes to the public data network. The address is specified in digits from 0 through 9. The network ID is the CDCNET network identifier of the X.25 network solution.

4. If the X.25 network solution will connect to foreign hosts, you must enter a DEFINE_X25_GW command to define the gateway between CDCNET and the foreign host.

   DEFINE_X25_GW GATEWAY_NAME=name TRUNK_NAME=list 1..32 of name

   Only the required parameters are shown above. Optional parameters define the protocol IDs and CDCNET titles for the gateway for both NOS and NOS/VE environments. Refer to the DEFINE_X25_GW command description in chapter 8 for more information on the parameters.

Example:

```
senc c='define_x25_trunk lim=1 port=1 trunk_name=TYMN1' s=ndi_1
senc c='define_x25_interface trunk_name=TYMN1 public_data_network=TYMNET..
twoway_range=0..32' s=ndi_1
senc c='define_x25_network trunk_name=TYMN1..
remote_dte_address=3075551212 network_name=TYMNET_NET1 network_id=1234'..
s=ndi_1
```

Commands:

DEFINE_X25_TRUNK
DEFINE_X25_INTERFACE
DEFINE_X25_NET
DEFINE_X25_GW (Optional; used if the X.25 trunk is also to operate as a gateway to non-CDNA networks)
START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once the network solution is configured. The network solution automatically starts after configuration unless you include the parameter START=FALSE on the DEFINE_X25_NET command.

Deleting a Network Solution or NP Interface

A network solution can be logically deleted. However, the network solution should not be deleted if it is the only link between a DI and the rest of the network. For example, if you logically delete the Ethernet network solution which is the only path from a TDI to the rest of the network, you cut off that TDI from the rest of the network. You will be unable to access the TDI by NETOU to reenable the network solution; the only way to redefine the network solution is to manually reset the TDI.

- Ethernet Network Solutions

  To delete an Ethernet network solution, follow this procedure.

  1. Stop traffic on the network solution by entering the STOP_NETWORK command.

  2. Cancel the network solution's definition by entering the CANCEL_ETHER_NET command. This command also cancels the underlying Ethernet trunk, so a separate CANCEL_ETHER_TRUNK command is not needed. However, when redefining an Ethernet network solution, you must define the Ethernet trunk because it was cancelled (see Redefining a Network Solution in this chapter).

     CANCEL_ETHER_NET NETWORK_NAME=network_name

     Provide the logical name of the network solution for the NETWORK_NAME parameter.

  Examples:

```
send_command c='stop_network network_name=engin_bldg_net',s=engin_tdi_1
send_command c='cancel_ether_net network_name=engin_bldg_net',s=engin_tdi_1
```

- Channel Network Solutions

  The procedures to delete a channel network solution on NOS are different from the procedures on NOS/VE.

  - NOS

    To logically delete a channel network solution on NOS, follow this procedure.

    1. Stop traffic to a NOS Network Products host by entering the STOP_NP_INTERFACE command. This command identifies the NOS Network Products interface to the NOS host.

       STOP_NP_INTERFACE INTERFACE_NAME=interface_name

       Provide the logical name of the interface assigned by the DEFINE_NP_INTEFACE configuration command for the INTERFACE_NAME parameter.

    2. Cancel the configuration of the NP interface with a CANCEL_NP_INTERFACE command.

       CANCEL_NP_INTERFACE INTERFACE_NAME = interface_name

       Provide the logical name of the interface assigned by the configuration command, DEFINE_NP_INTERFACE for the NETWORK_NAME parameter.

    3. Cancel the configuration of the channel trunk with a CANCEL_CHANNEL_TRUNK command.

       CANCEL_CHANNEL_TRUNK TRUNK_NAME = trunk_name

       Provide the logical name of the trunk assigned by the configuration command, DEFINE_CHANNEL_TRUNK for the TRUNK_NAME parameter.

    Examples:

    ```
    senc c='stop_np_interface in=cyber_109', s=mdi1
    senc c='cancel_np_interface in=cyber_109', s=mdi1
    senc c='cancel_channel_trunk tn=cyber_101_alt'
    ```

  - NOS/VE

    To logically delete a channel network solution on NOS/VE, follow this procedure.

    1. Stop traffic on the network solution that includes a NOS/VE host by entering the STOP_NETWORK command.

       STOP_NETWORK NETWORK_NAME=network_name

       Provide the logical name of the network solution for the NETWORK_NAME parameter.

    2. Cancel the configuration of the channel network and the underlying channel trunk definition by entering a CANCEL_CHANNEL_NET command.

       CANCEL_CHANNEL_NETNETWORK_NAME =network_name

       Provide the logical name of the network, assigned by the DEFINE_CHANNEL_NET configuration command for the NETWORK_NAME parameter.

3. Cancel the configuration of the channel by entering a CANCEL_CHANNEL_TRUNK command.

   CANCEL_CHANNEL_TRUNKTRUNK_NAME=trunk_name

   Provide the logical name of the trunk, assigned by the DEFINE_CHANNEL_TRUNK configuration command for the TRUNK_NAME parameter.

Examples:

```
send_command c='stop_network network_name=channel_net_1',s=ndi_1
send_command c='cancel_channel_net network_name=channel_net_1',s=ndi_1
send_command c='cancel_channel_trunk tn=cyber_101_alt'
```

- HDLC Network Solutions

To logically delete an HDLC network solution, follow this procedure.

1. Stop traffic on the network solution by entering the STOP_NETWORK command.

   STOP_NETWORK NETWORK_NAME=network_name

   Provide the logical name of the network solution for the NETWORK_NAME parameter.

2. Cancel the HDLC network solution by cancelling the logical definition of the HDLC network and the HDLC trunk by entering the CANCEL_HDLC_NET command. This also cancels the underlying trunk definition.

   CANCEL_HDLC_NET NETWORK_NAME=network_name

   Provide the logical name of the HDLC network for the NETWORK_NAME parameter.

Examples:

```
send_command c='stop_network network_name=tymnet_net_1',s=ndi_1
send_command c='cancel_hdlc_net network_name=menlo_park_network
```

- X.25 Network Solutions

To logically delete an X.25 network solution, follow this procedure.

1. Stop traffic on the network solution by entering the STOP_NETWORK command.

   STOP_NETWORK NETWORK_NAME=network_name

   Provide the logical name of the network solution for the NETWORK_NAME parameter.

2. Cancel the network solution's definition by entering the CANCEL_X25_NET command.

   CANCEL_X25_NET NETWORK_NAME=network_name

   Provide the logical name of the network solution for the NETWORK_NAME parameter.

3. Stop the X.25 Packet Level interface by entering the STOP_X.25_INTERFACE command.

   STOP_X.25_INTERFACE INTERFACE_NAME = interface_name

   Provide the logical name of the X.25 interface for the INTERFACE_NAME parameter.

4. If the X.25 interface that supports the network solution is also to be cancelled, enter the CANCEL_X25_INTERFACE command. If the X.25 interface has other active users, such as an X.25 gateway, do not cancel the X.25 interface.

   CANCEL_X25_INTERFACE INTERFACE_NAME=name

   Provide the logical name of the interface assigned by a DEFINE_X25_INTERFACE configuration command for the INTERFACE_NAME parameter.

5. If the logical definition of the trunk that supports the network solution is to be also cancelled, enter the CANCEL_X25_TRUNK command.

   CANCEL_X25_TRUNK TRUNK_NAME=trunk_name

   Provide the logical name of the trunk for the TRUNK_NAME parameter.

   If the X.25 interface remains, do not cancel the trunk.

Examples:

```
send_command c='stop_network network_name=tymnet_net_1',s=ndi_1
send_command c='cancel_x25_net network_name=tymnet_net_1',s=ndi_1
send_command c='stop_x25_interface network_name=tymnet_net_1',s=ndi_1
send_command c='cancel_x25_interface interface_name=tymnet_1',s=ndi_1
send_command c='cancel_x25_trunk trunk_name=tymnet_trunk_1',s=ndi_1
```

Commands:

Ethernet:

STOP_NETWORK
CANCEL_ETHER_NET

Channel:

NOS:

STOP_NP_INTERFACE
CANCEL_NP_INTERFACE
CANCEL_CHANNEL_TRUNK

NOS/VE:

STOP_NETWORK
CANCEL_CHANNEL_NET
CANCEL_CHANNEL_TRUNK

HDLC:

STOP_NETWORK
CANCEL_HDLC_NET

X.25:

STOP_NETWORK
CANCEL_X25_NET
STOP_X25_INTERFACE (Optional)
CANCEL_X25_INTERFACE (Optional)
CANCEL_X25_TRUNK (Optional)

Redefining a Network Solution or NP Interface to NOS

To redefine a network solution's logical definition, first cancel the current definition, then provide the values for the new definition. This subsection presents the sequence of commands required to redefine Ethernet, channel, X.25, and HDLC network solutions.

- Ethernet

  The CANCEL_ETHER_NET also cancels the underlying Ethernet trunk, so a separate CANCEL_ETHER_TRUNK command is not needed. However, when redefining an Ethernet network solution, you will have to define the Ethernet trunk, since it was cancelled.

  STOP_NETWORK
  CANCEL_ETHER_NET
  DEFINE_ETHER_TRUNK
  DEFINE_ETHER_NET
  START_NETWORK

- Channel:

  - NOS:

    STOP_NP_INTERFACE
    CANCEL_NP_INTERFACE (Optional)
    CANCEL_CHANNEL_TRUNK
    DEFINE_CHANNEL_TRUNK
    DEFINE_NP_INTERFACE (Optional)
    START_NETWORK

  - NOS/VE

    STOP_NETWORK
    CANCEL_CHANNEL_NET
    CANCEL_CHANNEL_TRUNK
    DEFINE_CHANNEL_TRUNK
    DEFINE_CHANNEL_NET
    START_NETWORK

- X.25

  STOP_NETWORK
  CANCEL_X25_NET
  STOP_X.25_INTERFACE
  CANCEL_X25_INTERFACE
  CANCEL_X25_GW (if applicable)
  CANCEL_X25_TRUNK
  DEFINE_X25_TRUNK
  DEFINE_X25_INTERFACE
  DEFINE_X25_NET
  DEFINE_X25_GW (if applicable)
  START_NETWORK

  If you only want to redefine the network solution, enter the following commands.

  STOP_NETWORK
  CANCEL_X25_NET
  DEFINE_X25_NET
  START_NETWORK

- HDLC:

  STOP_NETWORK
  CANCEL_HDLC_TRUNK
  DEFINE_HDLC_TRUNK
  DEFINE_HDLC_NET
  START_NETWORK

NOTE

The START_NETWORK command is required only if you do not want the network solution to automatically start once it is configured. (By default, it is started) This is set by the START parameter on the DEFINE_ETHER_NET, DEFINE_X25_NET, DEFINE_CHANNEL_NET and DEFINE_HDLC_NET commands.

Adding Terminal Devices

Network commands define the logical configuration of terminal devices, batch devices, I/O stations, and Network Transfer Facility (NTF) remote systems.

Terminal Devices

To add a terminal device to a DI's logical configuration, a terminal definition procedure (TDP) must be created that contains the DEFINE_TERMINAL_DEVICE command. If authorized to create TDPs for your site, refer to the CDCNET Configuration and Site Administration Guide for information about creating TDPs. If only site administrators are authorized to create TDPs at your site, notify the site administrator to create a TDP for a terminal, and provide the values for the parameters listed in the DEFINE_TERMINAL_DEVICE command description (see chapter 8).

Batch Devices, I/O stations, and NTF Remote Systems

Logical configuration of batch devices, I/O stations, and NTF remote systems is covered in the CDCNET Configuration and Site Administration Guide. Operation of batch devices is covered in the CDCNET Batch Device User Guide and the Remote Batch Facility (RBF) Reference manual. Refer to these manuals for detailed information on configuring and operating batch devices.

Batch I/O stations and individual devices are configured using terminal definition procedures (TDPs). TDPs contain commands to define the logical group of batch devices called an I/O station, to define parameters that apply to all the devices in the I/O station, and to define parameters that apply to the individual batch devices such as printers in the I/O station. The following commands are used in TDPs for I/O stations.

DEFINE_BATCH_DEVICE
DEFINE_I_O_STATION
DEFINE_NP_BATCH_STATION
DEFINE_TERMINAL_DEVICE
DEFINE_USER_I_O_STATION

NTF Remote Systems are configured using TDPs. The following commands are used in TDPs for NTF Remote Systems.

DEFINE_ACCESSIBLE_REMOTE_SYSTEM
DEFINE_BATCH_STREAM
DEFINE_REMOTE_SYSTEM

TDPs are created during network configuration, but they can be modified and new ones can be created. Refer to the CDCNET Configuration and Site Administration Guide for more information on creating and modifying TDPs and configuring I/O stations and NTF remote systems. TDPs are either executed automatically when the line connected to the I/O or remote system station becomes active or when a station operator executes the TDP using the DO command. For example, the following command executes a TDP named STATION1.

```
DO,STATION1
```

If a user were already connected to a host service, the network command character would have to be used with the DO command, as shown in the following example.

```
%DO,STATION1
```

Once batch I/O stations and their devices are active, you can perform operations such as starting and stopping devices as described in the CDCNET Batch Device User Guide and the NOS RBF Reference manual.

Controlling Gateways

This section describes how to control gateways. For this release of CDCNET, the X.25 gateway is the only gateway in which you can completely define, start, stop, and cancel in an online environment using NETOU commands. The complete set of commands is provided because it is assumed that for a typical site that uses X.25 services, starting and stopping X.25 services may be a daily activity.

Network Products Gateways

Commands can define and start the Network Products gateways, but these activities are usually done by including the commands in the DI configuration files (see DEFINE_NP_GW and DEFINE_NP_TERMINAL_GW command descriptions in chapter 8). The DEFINE_NP_GW command automatically starts the gateway when the command executes, so a start command is not currently supported. There are commands to start and cancel the Network Products interface (see START_NP_INTERFACE and CANCEL_NP_INTERFACE command descriptions in chapter 8).

The ADD_NP_GW_OUTCALL and DELETE_NP_GW_OUTCALL are used when a remote system must access applications residing on a NOS host. The outcall is from the perspective of the CDNET network; the call is going out of the CDCNET network. The add command provides the name (title) of the application through which remote systems can access applications residing on a NOS host. The delete command deletes the name (title) of the NOS gateway through which a remote system accessed applications residing on a NOS host. The name (title) is registered and maintained on a directory by the Directory Management Entity.

X.25 Gateways

The following commands are used to control access to foreign hosts connected to X.25 networks:

START_X25_INTERFACE
STOP_X25_INTERFACE
CANCEL_X25_INTERFACE
DEFINE_X25_INTERFACE
START_X25_GW
STOP_X25_GW
CANCEL_X25_GW
DEFINE_X25_GW
ADD_X25_GW_OUTCALL
DELETE_X25_GW_OUTCALL

The start, stop, cancel, and define commands control the X.25 interface. The start, stop, cancel, and define X.25 gateway (GW) commands control the X.25 gateway that provides access for NOS applications-to-applications on foreign systems connected to CDCNET by an X.25 public data network. The add and delete commands control the registration of the name (title) of the X.25 gateway in the Directory ME. The X.25 interface supports the X.25 gateway. When starting the X.25 gateway, first start the interface. When stopping the interface, you must first stop the X.25 gateway, and if an X.25 network solution is defined, the X.25 network solution. Stopping the X.25 interface also stops the trunk that supports the interface.

Figure 4-1 shows how the X.25 control commands are used to start and stop X.25 gateway services.

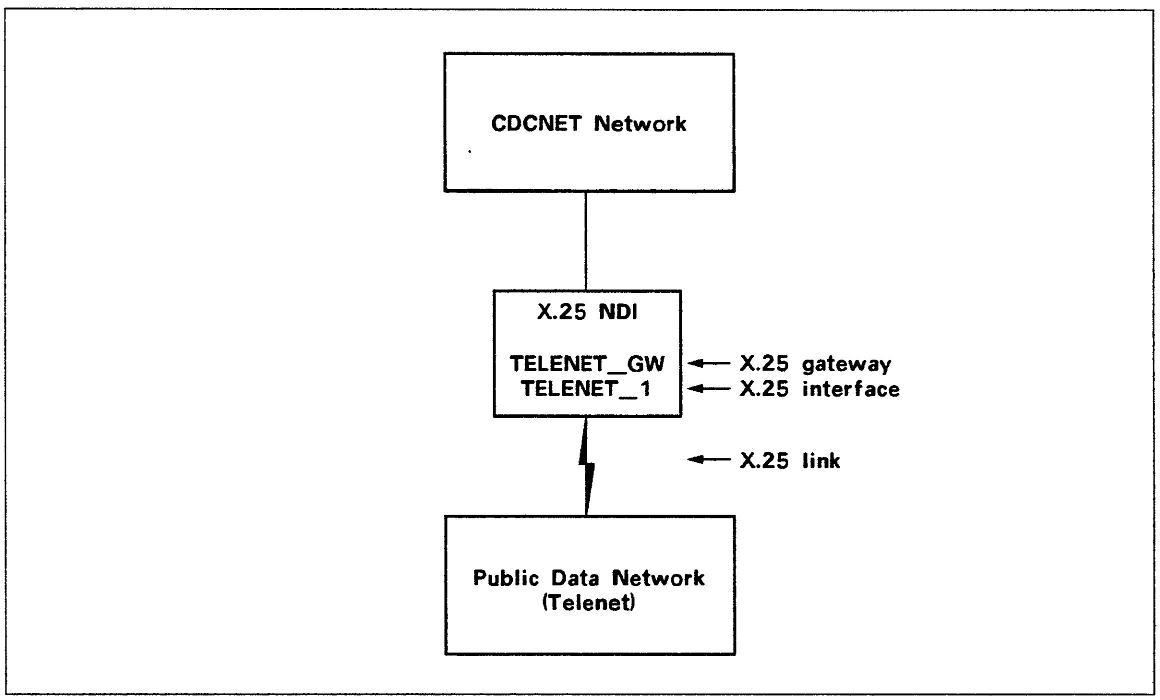


Figure 4-1. X.25 Gateway Example

Figure 4-1 shows an NDI connecting a CDCNET network with TELENET, a public data network, over an X.25 link. The X.25 interface to Telenet was defined during configuration in the NDI by the DEFINE_X25_INTERFACE, ADD_X25_GATEWAY_OUTCALL and DEFINE_X25_GW commands. The logical name for the X.25 interface is Telenet_1. The logical name for the X.25 gateway is Telenet_GW. CDCNET terminal users can access Telenet by starting and stopping X.25 gateway, Telenet_GW.

To start X.25 gateway services, the following commands are sent to the NDI.

```
start_x25_interface interface_name=telenet_1

start_x25_gateway gateway_name=telenet_gw

add_x25_gateway_outcall gateway_name=telenet_gw title=PTFS$TELENET
```

To stop X.25 gateway services, the following commands are sent to the NDI. The stop commands are sent in the opposite order of the start commands.

```
delete_x25_gateway_outcall gateway_name=telenet_gw title=PTFS$TELENET

stop_x25_gateway gateway_name=telenet_gw

stop_x25_interface interface_name=telenet_1
```

Logging and Alarm Control

This section describes activities for configuring and managing the CDCNET log and alarm message features. Network logging allows you to have a record of network activity in the form of log messages routed to a file on the host computer. Alarms are messages sent to your operations station that alert you to events in the network.

This section also refers to the utility that terminates the network log file on a NOS host, the Network Logfile Termination (NLTERM) Utility. If you are running CDCNET with a NOS host, you will have to use NLTERM periodically to close the current network log file and write the log messages to another permanent file. Directions for using NLTERM are in chapter 9.

Defining Log Messages to be Generated by a DI

The CDCNET logging structure consists of log message sources and log message recorders. Each DI is a log message source. The source provides log messages that describe the DI's activities. Each log message has a unique log message identifier. The complete list of these log messages and their identifiers is in the CDCNET Diagnostic Messages manual. In CDCNET networks connected to a NOS host, at least one DI in the network serves as a log message recorder. The recorder has access to permanent storage. Aided by a NOS CDCNET host application called the Network Log Server (NETLS), the recorder DI records the log messages from the source DIs into a host file known as the network log file.

In CDCNET networks connected to a NOS/VE host, the log message recording function is configured in the NOS/VE host. A log message recorder DI can not be defined in NOS/VE environments. Commands in the NOS/VE host START_UP file activate and deactivate the network logging function: ACTIVATE_NETWORK_LOG and DEACTIVATE_NETWORK_LOG. For more information on these commands, refer to the NOS/VE System Analyst Reference Set, Network Management.

There are network operations commands to configure and reconfigure the logging structure of your network. At each DI, there are lists maintained of what messages should be logged. The commands that affect logging sources allow you to define, change and cancel one or more log messages at each DI.

If you have logging sources defined in your network, you should have a logging recorder defined for the network, or a portion of the memory in the network's DIs will be used up by queued log messages generated by the DIs.

During network configuration, a default set of log message numbers are defined for each DI in the network with the DEFINE_SOURCE_LOG_GROUP command. These default messages are defined by commands in the DI configuration files created by the site administrator. Information on this activity is provided in the CDCNET Configuration and Site Administration Guide. You may add messages to this default set, but it is not recommended that you delete messages from the default set.

You can use the Network Performance Analyzer (NPA) Utility to look at log messages. Refer to the NPA manual for information.

Adding Log Messages to the Currently Defined List for Source DIs

1. Display the log messages that are currently logged at the source DIs using the DISPLAY_SOURCE_LOG_GROUP command.

2. Add or delete the messages you want to enable or disable using the CHANGE_SOURCE_LOG_GROUP command. Refer to the CDCNET Diagnostic Messages manual for message numbers.

Cancelling and Redefining Log Messages

You can also cancel and redefine the list of log messages to be generated at a DI by using the CANCEL_SOURCE_LOG_GROUP and DEFINE_SOURCE_LOG_GROUP commands.

CAUTION

It is recommended that you limit the number of log messages generated by a DI, since the messages are logged on the host disk space. If a large number of messages, particularly the entire set of log messages, are enabled for a DI, a significant amount of network traffic will be dedicated to transmitting log messages to the host. The log message feature may be useful for tracking problems or events in the network. However, enabling too many log messages can put constraints on DI and host memory.

Changing the Logging Recorder DI (NOS Only)

In this activity, you control which host is to record log messages. This procedure is performed only in CDCNET networks that are supported by NOS hosts. Address the commands only to MDIs/MTIs that provide for log recording.

You can use the CHANGE_RECORDER_LOG_GROUP command to directly change the priority of a log group rather than having to cancel and redefine the log group.

To cancel and redefine the recorder log group, follow this procedure:

1. Cancel the current log group to be recorded using the CANCEL_RECORDER_LOG_GROUP command.

2. Redefine the log group to be recorded using the DEFINE_RECORDER_LOG_GROUP command.

   NOTE

   If you cancel the recorder log group, you cancel the recording function for the entire catenet unless the log recording function is defined on multiple MDIs in the catenet. Cancelling and redefining a log recording function should be done only if you move the log message recording function from one DI to another.

Alarm Control

During network configuration, a default set of alarm message numbers are defined for each DI in the network by the DEFINE_SOURCE_ALARM_MESSAGE command. These alarms are sent to an alarm recorder. For NOS/VE operating systems, this is a host that executes the ACTIVATE_NETWORK_ALARMS command. Information on this activity is provided in the CDCNET Configuration and Site Administration Guide. You may add messages to this default set, but it is not recommended that you delete messages from the default set.

The initial set of DIs that report alarm messages to your operations terminal or console is all the DIs in the catenet. NOS/VE requires you to enter the ACTIVATE_ALARMS command in order to receive alarms from the DIs. NOS has no such requirement; the alarms activate by default on NOS.

Occasionally, you may choose to redefine the list of alarm messages and/or the set of DIs that report alarms to you. For example, if a DI is undergoing tests and generating many alarms, and the DI is being monitored by test personnel, you can shut off receipt of the alarms from that DI.

There are two main activities involved in alarm control.

- Defining alarm messages to be delivered from a DI.

  This activity allows you to add and delete alarm messages from the list of alarms which are to be reported to all operators in the network from a particular DI.

- Controlling your alarm environment (NOS Only)

  This activity allows you to control which DIs report alarms to you. You may temporarily shut off receipt of alarms from a DI at your operations terminal/console. Refer to the Session Control chapter in this manual for commands which control your operations alarm environment.

Defining Alarm Messages to be Generated by a DI

To initially define the set of alarm messages to be delivered from the source DI, use the DEFINE_SOURCE_ALARM_MESSAGE command. Refer to the CDCNET Diagnostic Messages manual for the message numbers.

CAUTION

It is recommended that you limit the number of alarm messages defined for a DI. If a large number of alarms are enabled, the amount of network traffic devoted to alarm message transmission will be increased. In addition, your operations station will be constantly receiving alarms. The alarm message feature may be useful for tracking problems or events in the network. However, enabling too many alarms can put constraints on available DI memory.

Controlling Alarm Environment

To redefine the set of messages to be delivered from the source DI as alarms, enter the following commands.

1. DISPLAY_SOURCE_ALARMS (To display alarm messages enabled).

2. CANCEL_SOURCE_ALARM_MESSAGE (To delete messages).

3. DEFINE_SOURCE_ALARM_MESSAGE (To add messages).

Provide the identification numbers for the messages you want the DI to send as alarms, surrounded by parentheses. Refer to the CDCNET Diagnostic Messages manual for the message numbers. To add alarm messages to the existing set, you can enter a DEFINE_SOURCE_ALARM_MESSAGE without having to cancel the existing set of messages.

NOTE

In order for the REQUEST_NETWORK_OPERATOR (REQNO) terminal user command to work, message number 168 must be enabled as an alarm.

Terminating and Archiving Network Log Files

CDC host computers provide logging capabilities to the CDCNET. Hosts maintain a network log file that receives log messages sent from DIs. Periodically, the current network log file must be terminated, and a new file to which new log messages will be written must be defined.

Terminating Network Log Files on NOS/VE

On NOS/VE, the network log file resides on file LOG in the $SYSTEM.CDCNET catalog. Individual sites can define the log file size limit, maximum number of log file cycles, and the interval at which a log file is terminated and analyzed, by specifying these values as parameters on the ACTIVATE_NETWORK_LOG command. This command can be entered by a network operator or be included in the NOS/VE host's file $SYSTEM.NETWORK_START_UP_COMMANDS and executed when NOS/VE is started. For more information on the ACTIVATE_NETWORK_LOG command, refer to the NOS/VE System Analyst Reference Set, Network Management.

Parameters used to define log file termination and processing on the ACTIVATE_NETWORK_LOG command include MAXIMUM_LOG_CYCLES, MAXIMUM_LOG_SIZE, INTERVAL and PROCESS_LOG_JOB.

MAXIMUM_LOG_CYCLES specifies the maximum number of log file cycles allowed. When this limit is reached, logging is suspended until one or more log file cycles are deleted. The default value is 999 cycles.

MAXIMUM_LOG_SIZE specifies the maximum size (in bytes) of the log file. When this file size is reached, the log file will be terminated, a file called PROCESS_LOG_JOB will be submitted as a batch job (see the following description of PROCESS_LOG_JOB), and a new log file cycle started. The default maximum file size for log files is the NOS/VE maximum file size (2^{48} bytes). If the keyword value NONE is specified for this parameter, the NOS/VE maximum file size is used.

INTERVAL establishes the time interval (in minutes) at which log files are to be terminated and processed and a new log file created. The default for this parameter is for no periodic processing; a log file is terminated when it reaches a certain file size, rather than when a time period elapses.

The PROCESS_LOG_JOB is a file containing a batch job that is automatically run each time a network log file is terminated. Typical functions which could be performed by this job include running Network Performance Analyzer Commands such as REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR), and purging and archiving log files. The complete file reference for this file is $SYSTEM.CDCNET.VERSION_INDEPENDENT.PROCESS_LOG_JOB. The actual contents of this file are site-definable. Control Data provides a sample batch job in NOS/VE / CDCNET release materials which can be used initially and modified as needed at your site. Refer to the file for its contents.

Terminating Network Log Files on NOS

On NOS, the network log file is a NOS direct access permanent file under user name SYSTEMX. The network log file is not automatically terminated. You must use the Network Log File Termination Utility (NLTERM) to terminate log files. The Network Logfile List Utility (NLLIST) provides a list of all terminated network log files that have not been purged. The function of NLLIST is also performed by an NLTERM subcommand called LIST.

The directions for using NLTERM are in chapter 9 of this manual.

NLTERM can be run as part of a daily system closedown process submitted as a batch job.

Archiving Network Log Files

Archiving log files that have been terminated is an additional log file management step which may be appropriate for your site, depending on your site's network configuration, how much log traffic is generated, and how large your log files are.

Once log files are terminated, they should be reformatted by the NPA procedure REFORMAT_CDCNET_LOG_FILE (REFCLF). Reformatted log files may be moved to tape and deleted from disk. Databases generated from CDCNET log files may also be archived using the ARCHIVE_NPA_DATA_BASE (ARCNDB) command. Refer to the Network Performance Analyzer manual for information on these activties.

Statistics Control

CDCNET statistics are numerical indicators of network performance. They are counts of data traffic and various events detected by the CDCNET communications software. Some examples of statistics include the number of messages or characters transmitted or received per line or DI and the number of errors encountered during a sampling period. Statistics may be used to determine how the network is performing and to identify potential or real problems such as failing software processes or communication bottlenecks on lines and network solutions.

You may gather CDCNET statistics using statistics control commands. These commands start and stop the collection and reporting of statistics for the following network components: network solutions, communication lines, and software processes (such as the file access management function, log message recording, and gateways). There are start and stop commands for each of the three types of components for which you may gather statistics.

Statistics collection is started for the three types of statistics that may be collected (line, network, and process) by start metrics commands. Once started, statistics are gathered over a collection period called a report interval. The report interval is set by a parameter on each START command. This interval may differ between the components you are sampling. Collection of statistics is continuous; when one interval ends, another starts. At the end of a report interval, the statistics gathered during the report interval are reported by a log message, which is placed in the network log file, and a new report interval begins.

You may stop the collection and reporting of statistics by the stop metrics commands. These commands may be entered either before or after you obtain the statistics results (refer to Obtaining Statistics Results).

The appropriate log messages must be enabled for you to receive statistics information. The default set of log messages enabled by CDCNET includes the appropriate log messages providing statistical information.

Statistics Groups

There are three levels of statistics that are collected: summary, expanded, and debug statistics.

Summary statistics provide an overview of the operation of a line, network solution, or software process. Examples include the number of messages received and characters transmitted. In most cases, summary statistics will provide sufficient statistical information about a component's performance.

Expanded statistics are a refinement of summary statistics. Examples include response times for a terminal user, number of messages processed for each user, and distribution of size of messages transmitted and received by a software component. Expanded statistics are useful in cases where a service is being provided for an individual user through a connection, because they can give more specific information about the connection and how the service is performing using a particular connection. In contrast, summary statistics provide an overview of how the service is working for all users. Not every component supports expanded statistics.

Debug statistics are a further refinement of statistics and include information that can be used to debug software components. Examples include the amount of global memory used and memory addresses involved. Not every statistic type has expanded and debug levels; only process statistics have the debug statistics group.

An example of statistics groups can be seen in the statistics that may be gathered for the software process called XNS Transport. Summary XNS Transport statistics report the number of transport protocol units received and transmitted. Expanded statistics report the number of connections opened and closed for each service access point (SAP), the number of data units received and transmitted per connection, the number retransmitted per connection, and the number of duplicate data units received per connection.

Statistics levels are not hierarchical. You can start collection of expanded or debug statistics without also starting summary statistics collection. The default group level for all START commands is summary statistics. The default for all STOP metrics commands is to stop all statistics groups. When you stop statistics collection and reporting by specifying groups, any statistics groups not specified in the command remain in effect. However, if you send a start metrics command and have all statistics groups reporting, and later stop statistics without specifying all groups, any groups not specified will continue to be collected and reported.

Starting and Stopping Statistics

1. Start the statistics using one or all of the following start metrics commands (send the commands within SEND_COMMAND).

   START_LINE_METRICS
   START_NETWORK_METRICS
   START_PROCESS_METRICS

```
senc c='start_line_metrics line_name=line31..
report_interval=300',s=west_tdi

senc c='start_network_metrics network_name=ether1..
report_interval=300',s=mdi1

senc c='start_process_metrics process=xns_transport..
report_interval=300',s=tdi_3
```

2. Enter one or all of the following stop metrics commands either before or after obtaining the statistics results.

   STOP_LINE_METRICS
   STOP_NETWORK_METRICS
   STOP_PROCESS_METRICS

```
senc c='stop_line_metrics line_name=line31',s=west_tdi
senc c='stop_network_metrics network_name=ether1',s=mdi1
senc c='stop_process_metrics process=xns_transport',s=tdi_3
```

   For complete descriptions of the above commands refer to each command's description in the Network Commands chapter.

Obtaining Statistics Results

CDCNET statistics are reported by log messages, which are written to the CDCNET network log file on the host computer. You can display the statistics by defining the log messages as alarm messages, using the DEFINE_SOURCE_ALARM_MESSAGE command. This is discussed below.

Statistics can be obtained by reformatting the CDCNET log file containing the statistics messages using Network Performance Analyzer (NPA) commands. This manual briefly describes what to do to receive statistics reports; refer to the NPA manual for complete information on NPA reports and the commands used to generate them.

To reformat the CDCNET network log file to obtain statistical reports, use an NPA command called REFORMAT_CDCNET_LOG_FILE (REFCLF). Refer to the NPA manual for the REFCLF command format and parameters. REFCLF reorganizes the network log file (a chronological list of all log messages generated by the network's DIs), and builds files of various types of log messages called databases. NPA has standard database types and names. Each database contains a certain type of log message, such as log messages for a DI's CPU and memory use, or messages relating to terminal and connection performance. These databases are used to develop statistics reports.

Statistics reports are created from the NPA databases using another NPA command, CREATE_CDCNET_ANALYSIS_REPORT (CRECAR). Refer to the Network Performance Analyzer (NPA) manual for the CRECAR command format and parameters.

Log file reformatting and report generation (by the NPA commands REFCLF and CRECAR) may be done by running a routine batch job when the network and operating system are being shut down or started-up.

While statistics are being reported, you can monitor statistics messages at your operations station by enabling the statistics messages as alarms. Use the DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM) command to enable the messages as alarms. The message numbers to enable for line, network, and process metrics are shown in the table 4-1.

Table 4-1. Statistics Commands and Message Numbers

Command	Message Number
START_LINE_METRICS	166
START_NETWORK_METRICS	639, 665, 562
START_PROCESS_METRICS	299, 737

Running Network Performance Analyzer (NPA) Procedures

Besides REFORMAT_CDCNET_LOG_FILE and CREATE_CDCNET_ANALYSIS_REPORT, there are other NPA-related activities you may have to perform using NPA commands.

- Archiving NPA databases (ARCNDB)
- Reloading NPA databases (RELNDB)
- Explaining CDCNET Log Messages (EXPCLM)
- Editing CDCNET Log Messages (EDICLM)

The ARCNDB command removes records from the NPA databases and puts the information in an archive file. The RELNDB command reloads records from an archive file and merges these records into the existing NPA database. The EXPCLM command provides information on the CDCNET log message you specify. The EDICLM command allows you to create, add to, or change the site information section of a CDCNET log message.

To learn how to use these and other NPA commands, refer to the Network Performance Analyzer manual.

Stopping, Resetting, and Dumping a DI

You may have to stop a DI while the network is running. A DI is stopped through NETOU by using the KILL_SYSTEM command. This command shuts off the system clock in the DI, which forces the DI to stop immediately. After a few seconds, the DI resets and reloads, and optionally dumps its memory to a file.

If you have made several changes to a DI's logical configuration using the operations interface, and want to return to the standard definition of the DI, you can reset the DI using the following procedure. This will cause the originally defined configuration file for the DI to be read and executed, which returns the DI's definition to its standard form.

To stop a DI:

1. Notify all active users that they will be disconnected from CDCNET services (see Sending Messages to Terminal Users).

2. Send the KILL_SYSTEM command to the DI that is to be stopped.

    KILL_SYSTEM DUMP=YES or NO

    If you want a dump of the DI's memory to occur, specify YES. If you do not want a dump to occur, specify NO.

Loading and Unloading Software

The DI has a software component called the online loader, which can load new software while the DI is running. The online loader can also unload software from the DI to make more room in the DI. For example, the online loader may unload software that is currently not being used to make room for more connections or more space for building tables.

Two commands, LOAD_MODULE and UNLOAD_MODULE, allow you to do the work of the online loader. The LOAD_MODULE command immediately loads software into the DI. If a value of YES in entered in the RETAIN parameter of this command, the module will not be unloaded to recover system memory resources, unless an UNLOAD_MODULE command is used.

The UNLOAD_MODULE command is used to remove a software module from a DI. For example, you may want to move a service from one DI to another, or unload a command processor or diagnostic that was loaded by the LOAD_MODULE command. The UNLOAD_MODULE command allows a module to be unloaded. It clears the retain flag from the module, so that when the module is no longer used it can be unloaded if memory is needed. Using UNLOAD_MODULE does not guarantee that a new version of a module will be loaded the next time the module is used.

Refer to the command descriptions for LOAD_MODULE and UNLOAD_MODULE in chapter 8 for more information on these commands.

NOTE

It is recommended that you use these commands under the supervision of a CDCNET analyst.

TCP/IP Networks

This section provides instructions for network operations activities on TCP/IP networks.

TCP/IP Gateways

Commands can define and start TCP/IP gateways, but these activities are usually done by including the commands in the DI configuration files (see DEFINE_USER_TELNET_GW and DEFINE_SERVER_TELNET_GW command descriptions in chapter 8). The DEFINE_USER_TELNET_GW command automatically starts the user gateway when the command executes. The DEFINE_SERVER_TELNET_GW command automatically starts the server gateway when the command executes.

The following examples illustrate how to cancel and redefine USER_TELNET and SERVER_TELNET gateways:

```
senc c='stop_user_telnet_gw gateway_name=gw_to_vax'
senc c='cancel_user_telnet_gw gateway_name=gw_to_vax'
senc c='define_user_telnet_gw gateway_name=gw_to_vax,..
ip_address=(128,5,0,3),..
title=vax_86'
senc c='start_user_telnet_gw gateway_name=gateway_to_vax'


senc c='stop_server_telnet_gw gateway_name=gw_to_cyber'
senc c='cancel_server_telnet_gw gateway_name=gw_to_cyber'
senc c='define_server_telnet_gw gateway_name=gw_to_cyber,..
ip_address=(128,5,0,2),..
title=VE_990'
senc c='start_user_telnet_gw gateway_name=gateway_to_cyber'
```

IP Host

The following example shows how to cancel and redefine an IP_Host:

```
senc c='cancel_ip_host ip_address=(128,5,0,3)'
senc c='define_ip_host ip_address=(128,5,0,3),..
                       host_type= ip_host,..
                       system_id=(070701(16),009ECB(16))
```

Troubleshooting 5

Troubleshooting 5

As a network operator, you must make some basic troubleshooting decisions during problem situations. This chapter discusses troubleshooting of CDCNET problems at the network operator level. It focuses on the commands and activities available to you through the Network Operator Utility (NETOU) and contains guidelines you should follow when troubleshooting. More extensive coverage of hardware troubleshooting, including onboard and online diagnostic tests, is in the CDCNET Network Troubleshooting Guide. Analysis of software problems is covered in the CDCNET Network Analysis manual and the Network Performance Analyzer (NPA) manual. If problems occur that cannot be remedied by network operations commands, or if NETOU is unavailable due to the problem, your site should refer to these manuals for further troubleshooting information.

Troubleshooting Guidelines

When monitoring and directing the network through NETOU, follow these guidelines.

- Keep aware of alarms, which can point to developing problems and emergencies. If you have been away from your operations station for some time, check the alarm history (see chapter 3).
- Check hardware and software status for DIs using the display status commands.

If the NPA reports are run on a regular basis at your site, review the reports.

When problems occur, follow these guidelines.

- When you get calls about problems from users, first identify the problem, even if the identification is a broad one or a symptom common to several conditions.
- Next, decide whether you want to try to solve the problem yourself by sending commands, or if you want to report the problem as-is to site maintenance personnel. You can also report the problem to CDC by writing a Programming System Report (PSR) about the problem, resetting and dumping the DI with the problem, and sending the PSR and dump to CDC. If you send any commands to a DI, you should still write a PSR, but note any commands sent and any other actions you took to solve the problem. Refer to the CDCNET Site Administration and Configuration manual for information about writing a PSR.
- The actions you take should reflect the apparent severity and extent of the problem. If the problem seems to be related to one line or board, the rest of the network should be able to remain operational while the problem is being resolved. For example, if a single terminal is hung, you should not reset the DI connected to the terminal unless you have tried every other option.
- Check for problems that seem obvious, such as loose line and Ethernet cable connections, inappropriate hardware hookups, unplugged DIs and terminals, and options set at terminals that conflict with the CDCNET configuration for the terminal.

- Gather information for the system analyst's use.

  Send the display status commands to the DI or DIs that seem to be experiencing problems. The command responses should be routed both to your display and to a file (see chapter 3). The command response file can be reviewed by the analyst, if necessary.

  Determine whether and when to initiate a dump for analysis under the Device Interface Dump Analyzer, as documented in the CDCNET Network Analysis manual.

- Check the configuration procedures and files for the network. Commands and parameter values could be misspelled, or procedures and files may have been purged from the catalog or their permits changed for configuration. On NOS/VE, check the procedures in $SYSTEM.CDCNET.SITE_CONTROLLED catalog. On NOS, check the procedures and files in catalog for user name NETADMN.

- If a CDCNET terminal user reports a problem, try to get an example of the problem. While the user is reporting the problem, review the configuration of the DI to which the user is connected, and the line and terminal configuration.

- If users cannot connect to services after entering a CREATE_CONNECTION command, ask the users which title or titles they are specifying, and verify these titles with the site administrator. Users may not have been notified of the titles a user specifies to connect to host services using the CREATE_CONNECTION command. It is the CDCNET site administrator's responsibility to distribute these names to users. If changes to the titles occur, it is also the site administrator's responsibility to distribute the new names to users.

- A problem that keeps a user from accessing the network could be due to several things, but sometimes the causes are hidden in the software, which makes the problem harder to detect and correct. This could happen to equipment that has many options, such as communication lines from terminals to DIs and switch settings on terminals and modems. The problem a user is having may not be due to faulty I/O boards or bad lines; it could be due to another user changing the terminal software options and not notifying other users. Be sure to check out these possibilities at your site.

- Line connection failures could be due to flow control problems. Check lines to see if users are trying to make the lines run at speeds the lines cannot support. For example, a line that has a defined line speed of 9,600 bits per second should not run at 19,200 bits per second, or flow control problems will result.

- If you cannot determine and isolate the problem yourself, refer the problem to your site analyst or customer engineer. Provide the analyst or customer engineer with all the information you have gathered about the problem.

Problem Reporting

Some of the problems you encounter with NETOU or other CDCNET software may require that you report the problem to CDC. To do this, submit a Programming System Report (PSR) either by sending a hard copy PSR form or by using the online SOLVER database. The CDCNET Configuration and Site Administration Guide has an appendix on problem reporting and writing an effective PSR.

Diagnostics

The online and inline diagnostics tests should be run on a DI as a last resort, if there seems to be enough evidence present to do so, and if none of the other operations commands you enter isolate the problem. Online diagnostics tests require exclusive access to and control of the device being tested. Inline diagnostics tests share access to and control of the device being tested with nondiagnostic software. Try to determine whether the problem is confined to one user or port, or if it affects all users of the DI. To run diagnostics, you have to stop the device you are going to test, and this shuts off users from the network. There are descriptions of the commands used to start, stop and display online diagnostics tests in chapter 8. There is also a description of a command used to monitor the progress and outcome of any diagnostic tests you run. Refer to the Network Troubleshooting Guide for more information on online, inline, and onboard diagnostics.

Commands Used for Troubleshooting

This section lists the commands you will most likely use when isolating and solving network problems. Before you use any of these commands, you should generate or review reports on the network using the Network Performance Analyzer (NPA).

Information-Gathering Commands

These commands are the most common commands you will use. They display the operational status and current configuration of network components. If these commands do not immediately pinpoint the problem, the information they provide can be used to help further isolate the problem. If you cannot solve the problem yourself, you should pass the information provided by these commands (plus any NPA reports you have) on to the next level of support. Information gathering commands include:

- Display status commands.
- Display configuration commands.
- Statistics Commands.
- Display terminal and connection attributes commands.

NOTE

When you enter commands other than the DISPLAY commands such as START, STOP and DEFINE commands during a problem situation, you begin to alter the problem situation. If you enter any commands other than information-gathering commands and take steps to solve the problem yourself, you should write a PSR and note the commands that you have entered.

Start and Stop Traffic Commands

Use these commands to start a line or network solution that appears to be stopped, or to stop a line that is experiencing intermittent problems and will be tested. These commands can be used with the configuration commands to stop and reconfigure lines and network solutions. Start and stop traffic commands include:

- START_LINE and STOP_LINE
- START_NETWORK and STOP_NETWORK

KILL_SYSTEM Command

The KILL_SYSTEM command may be necessary in some problem cases. It will immediately reset a DI, and optionally dump the DI's memory. If you select the dump option on the command, a permanent file to receive the dump is automatically created. On NOS/VE, the dump file is located in the $SYSTEM.CDCNET.DUMP.SYSTEM_ sssssssssss catalog, where sssssssssss is the system ID of the DI. On NOS, the dump file is located in the user name NETOPS, and an entry for the file is made in the CDCNET file directory, NETDIR. Find the dump file with the command CATLIST or NETFM from user name NETADMN. Attach the file from NETADMN. The dump can then be examined by the DI dump analyzer program. Refer to the CDCNET Network Analysis manual for instructions on using the DI dump analyzer.

The appendix on problem reporting in the CDCNET Configuration and Site Administration Guide has more information on accessing and handling dump files from DIs and from host applications such as NAM or NAM/VE that support CDCNET.

Configuration Commands

These commands can be used if your analysis of the problem indicates that a configuration change may be necessary. You can change the configuration of DI software while keeping the network operational by sending the configuration commands to DIs through NETOU. However, configuration changes to DI software made through NETOU are temporary. The changes are discarded when the DI reloads, at which time the previous configuration is reestablished. To make permanent configuration changes, you have to change the configuration procedures for the DI. Refer to the CDCNET Configuration and Site Administration Guide for instructions on changing configuration procedures and files.

Using configuration commands, you can change the configuration of:

- Log messages and alarms sent from a DI.
- DI operating system tuning parameters that control memory management and buffer allocation (refer to Advanced Configuration Concepts chapter in CDCNET Configuration and Site Administration Guide).
- Communication lines connected to a DI.
- Network solutions connected to a DI.
- Terminal devices and their attributes.

Configuration commands entered through NETOU cannot be used to change the configuration for DIs that connect to NOS systems, the Network Products interactive terminal gateway that allow interactive terminal users to connect to NOS. This includes defining the title or titles that terminal users specify when connecting to NOS through CDCNET.

Configuration commands entered through NETOU can not be used to change the configuration of I/O stations, batch devices and their attributes.

It is important to check the configuration commands in configuration procedures if network problems occur. Check for misspelled commands and parameter values in configuration commands.

Examples of Troubleshooting Process

The following section presents two examples of network problems, and actions that a network operator takes to resolve those problems.

The problems described have different levels of severity. The actions taken reflect the severity of the problem and the number of users affected.

Example 1: Hung Lines on 1 LIM

In this example, a CDCNET network operator gets a call from a terminal user who is experiencing a hung line. The line is connected to a TDI supporting 32 lines. Two other lines on the same LIM are experiencing hung lines. The operator performs the following activities.

1. Has the terminal user enter the XON sequence (Break key plus CTRL Q).

2. Has the terminal user try to reactivate the line by turning off the terminal and turning it back on. If autorecognition occurs, the line is reactivated.

3. If steps 1 and 2 don't reactivate the line, the operator sends the DISPLAY_LINE_STATUS command to the TDI being examined to see the operational state of the line.

4. At this point, the operator decides whether to reset the TDI and submit a PSR and a dump of the TDI, or to try to stop and start the line using network commands.

   If the operator chooses to submit a PSR, the TDI is reset, and the operator's troubleshooting activities end here. The operator sends the dump and the PSR to CDC, and documents the steps taken before dumping the TDI.

   If the operator wants to get the line started again by sending network commands, the operator sends the following commands to the TDI.

   STOP_LINE
   DISPLAY_LINE_STATUS (this command is sent frequently throughout the troubleshooting process)
   START_LINE

5. If the STOP_LINE and START_LINE commands do not restart the line, the operator sends the following commands to the TDI.

   STOP_LINE
   CANCEL_LINE
   DEFINE_LINE

6. If redefining the line does not solve the problem, the operator checks whether other terminal users connected to the TDI are experiencing the same line hang problem. The operator runs the LIM online diagnostic test on the LIM supporting the lines, and displays the status of the LIM test using the DISPLAY_TEST_STATUS command.

7. Refer to the CDCNET Troubleshooting Manual.

Example 2: Hung Lines on Several LIMs

In this example, a TDI's lines are hanging, and the problem affects more than one LIM in the TDI. The operator performs the following activities.

1. Enters the DISPLAY_DI_SYSTEM_STATUS command. The operator checks the display to see if buffer and memory congestion is occurring.

2. If users across LIMs are affected, the TDI must be reset. The operator enters the KILL_SYSTEM command with the DUMP parameter set to YES.

Part 2: Reference

Command Description Format

The commands in all chapters of the Reference part of this manual are described using the following format:

- The command name followed by its valid abbreviation, if any, in parentheses.

- A brief description of the command's purpose.

- Command format. The command name is shown on the first line, with any plural forms or alternate spellings in parentheses. The command name is followed by parameters and the value list allowed for each parameter. Each command parameter is shown on a separate line in the order in which they must be entered if specified in positional format. The parameters may be entered in any order if they are specified in non-positional format. Required parameters appear in **boldface**. Optional parameters are in *italics*. The value list for parameters adheres to types defined in the System Command Language Definition manual. Refer to this manual for definitions of each of these terms.

- Description of each parameter, including its full name, its abbreviation, and possible values. If a parameter is optional, the default value that is assumed if you omit the parameter is specified.

- Remarks, including restrictions, rules, references to other manuals, or other special information about the command.

- Responses to the command. All responses except for informative responses begin with a severity level indicator, which may be one of the following:

  --WARNING--
  --ERROR--
  --FATAL--

  If a response does not display a severity level, it is an informative response that indicates successful command completion. Command responses are also listed and explained in the CDCNET Diagnostic Messages handbook.

- Example command and response. Where necessary, displays and responses are explained.

NOS/VE Session Control Commands 6

NOS/VE Session Control Commands 6

This chapter provides complete descriptions of all session control commands for NOS/VE-based network operations environments. Command descriptions are in alphabetical order. These are all operations session control commands and do not have to be contained within a SEND_COMMAND command.

The format for the command descriptions in this chapter is located on the divider page for part 2 of this manual.

ACTIVATE_ALARMS (ACTA) (NOS/VE Version)

Purpose Initiates receipt of alarms from DIs. This command must be entered after invoking NETOU to allow alarms to be reported to you.

Format **ACTIVATE_ALARMS**
GROUPS = list of name
OUTPUT = file
STATUS = status variable

Parameters *GROUPS* or *GROUP (G)*

Specifies the names of the alarm groups for which alarms are to be collected. Default value is CATENET. In this release of CDCNET, CATENET is the only value accepted for this parameter.

OUTPUT (O)

Specifies the file to receive the alarm messages. Default value is $OUTPUT.

STATUS

See basic status concept for NOS/VE SCL.

Responses --ERROR-- Alarms already active.

Remarks To ensure that alarms are activated each time you log in to NOS/VE and access NETOU, include this command in your user prolog.

Examples

```
activate_alarms

NOU/
```

DEACTIVATE_ALARMS (DEAA) (NOS/VE Version)

Purpose Terminates receipt of alarms from CDCNET DIs.

Format **DEACTIVATE_ALARMS**
STATUS = status variable

Parameters *STATUS*

See basic status concept for NOS/VE SCL.

Responses Alarms deactivated.

--ERROR-- Alarms not active.

Examples

```
deactivate_alarms

Alarms deactivated.
```

DISPLAY_COMMAND_LIST (DISCL)

Purpose Displays a list of network commands for which you are validated. The commands are arranged in alphabetical order. Only the long form of the command is returned. The HELP and DISPLAY_COMMAND_LIST_ENTRY commands are aliases for this command.

Format **DISPLAY_COMMAND_LIST**

Responses <Alphabetical list of network commands. See example.>

Examples

```
display_command_list

add_np_gw_outcall                        add_x25_gw_outcall
       .                                        .
       .                                        .
       .
unload_module                            write_terminal_message
```

QUIT (QUI) (NOS/VE Version)

Purpose Terminates the Network Operator Utility (NETOU) session. Once the QUIT command executes, NETOU commands will not be valid during a NOS/VE command entry session.

Format **QUIT**

Examples

```
quit
```

SEND_COMMAND (SENC) (NOS/VE Version)

Purpose Sends a CDCNET command to a DI or list of DIs.

Format

SEND_COMMAND
COMMAND = string
SYSTEM = name
OUTPUT = file name
STATUS = status_variable

Parameters

COMMAND (C)

The network operations command to be sent to the specified DI. Enter the command as a string value enclosed by apostrophes ('). You may use the abbreviated form of the command. If the command you are sending contains a string value (such as WRITE_TERMINAL_MESSAGE), you must use two consecutive apostrophes at the beginning and end of the string in order for the enclosed string to be recognized (see examples). You cannot substitute the quotation mark character for two apostrophes.

SYSTEM (S)

The logical or physical DI name or list of DI names to which the command is to be sent. If a CDCNET command is sent to more than one CDCNET system, a response must be received from each system for the command to complete.

OUTPUT (O)

The file to which a normal command response will be written. Default value is $OUTPUT. See the command response concept for NOS/VE-based CDCNET operations environments.

STATUS

See basic status concepts for NOS/VE System Command Language in the NOS/VE SCL Language Definition manual.

Examples

```
send_command command='display_hardware_status',system=mdi83
```

```
send_command c= 'write_terminal_message,..
m=''Engineering''''s network will be down until 10:00''',..
s=tdi1
```

NOS Session Control Commands 7

NOS Session Control Commands 7

This chapter provides complete descriptions of all session control commands for NOS-based network operations environments. Command descriptions are in alphabetical order. These are all operations session control commands and do not have to be contained within a SEND_COMMAND command.

The format for the command descriptions in this chapter is located on the divider page for part 2 of this manual.

ACTIVATE_ALARMS (ACTA) (NOS Version)

Purpose Initiates receipt of alarms from DIs.

Format **ACTIVATE_ALARMS**

GROUPS = list of <name>
OUTPUT = <file>

Parameters *GROUP (G)*

Specifies the names of the alarm groups for which alarms are to be collected. Default value is CATENET, which specifies all the DIs in the catenet.

OUTPUT (O)

The file which will receive the alarm messages. Default value is $OUTPUT.

Responses Alarms activated.

--ERROR-- Alarms already active.

Examples

```
activate_alarms

Alarms activated.
```

CHANGE_ALARM_ENVIRONMENT (CHAAE)

Purpose

Changes the list of DIs from which you receive alarms. You may shut off or again turn on the receipt of alarms from DIs. Use of this command does not affect alarms received by other network operators, if your network has more than one network operator.

This command can also change the list of DI communities from which you receive alarms. The community parameters are supported for this release. However, since the community feature is not supported until later releases, the parameters have only one allowed value, CATENET. If you disable receipt of alarms from a specific system, then you will receive no alarms from that system, regardless of the communities to which the system belongs. If you disable receipt of alarms from a specific community, however, you may receive alarms from any system that belongs to both the disabled community and some other community not disabled. Disabling alarms by system takes precedence over disabling alarms by community.

Format

CHANGE_ALARM_ENVIRONMENT
DISABLE_SYSTEM = list 1..15 of name
ENABLE_SYSTEM = list 1..15 of name
DISABLE_COMMUNITY = list 1..15 of name
ENABLE_COMMUNITY = list 1..15 of name

Parameters

DISABLE_SYSTEM (DS)

Name or names of DI or DIs for which receipt of alarms by the network operator is to be shut off. Entry of a name already disabled is permitted.

ENABLE_SYSTEM (ES)

Name or names of DI or DIs for which receipt of alarms by the network operator is to be turned back on. Entry of a name already enabled is permitted.

DISABLE_COMMUNITY

The community title or titles from which receipt of alarms is to be disabled. Entry of a title already disabled is permitted. For this release of CDCNET, the only allowed value for this parameter is CATENET, which specifies all the DIs in the catenet.

ENABLE_COMMUNITY

The community title or titles from which receipt of alarms is to be enabled. Entry of a title already enabled is permitted. For this release of CDCNET, the only allowed value for this parameter is CATENET, which specifies all the DIs in the catenet.

Responses

Alarm environment updated.

The following responses are not supported in this release of CDCNET.

--ERROR-- Community <name> is not in the operator's domain of control.

--ERROR-- System <name> is not in the operator's domain of control.

Examples

```
change_alarm_environment ds=engin_bld_tdi

Alarm environment updated.
```

DEACTIVATE_ALARMS (DEAA) (NOS Version)

Purpose Terminates receipt of alarms from CDCNET DIs.

Format **DEACTIVATE_ALARMS**

Parameters None.

Responses Alarms deactivated.

--ERROR-- Alarms not active.

Examples

```
deactivate_alarms

Alarms deactivated.
```

DISPLAY_ALARM_ENVIRONMENT (DISAE)

Purpose Displays the list of DI communities with your operations domain of control from which receipt of alarms is enabled or disabled. This command also lists the DIs from which alarms are disabled. For this release of CDCNET, the only community that is displayed is CATENET, and the only domain of control supported is the catenet.

Format **DISPLAY_ALARM_ENVIRONMENT**

Responses Alarm Environment
(See example.)

Examples

```
display_alarm_environment

Alarm Environment
  Community            Alarm Status
CATENET                Enabled

  Disabled Systems
-None-
```

DISPLAY_ALARM_HISTORY (DISAH)

Purpose Displays alarms received at your operations station in chronological order since the start of your command session. The limit for the display list is 50 display lines. If you receive more than 50 display lines, then new display lines replace the oldest alarms on the display. (Because there is a blank line between each alarm, you may see only 34 non-blank lines of text.)

Format **DISPLAY_ALARM_HISTORY**
DISPLAY_OPTION = keyword value

Parameters *DISPLAY_OPTION (DO)*

Specifies how many alarms will be displayed. You can display all alarms received since the last DISAH command was entered (up to the history limit), you can display the last page of alarms received, or you can display all alarms in the buffer.

Keyword Value	Description
LAST	Displays all alarms received since the last DISAH command was entered.
PAGE	Displays last page of alarms received.
ALL	Displays all alarms that are in the buffer, which has a limited buffer size.

Default is LAST.

Responses

```
ALARM HISTORY REPORT
****** ALARM FROM
```

<name>

<List of alarms. See example.>

No new alarms received since last DISPLAY_ALARM_HISTORY.

Examples

```
display_alarm_history
            ALARM HISTORY REPORT

****** ALARM FROM MTI_83             85/10/10  13.38.51      619
--ERROR--   Line: LINE31 down, connection timer expired

****** ALARM FROM MTI_83             85/10/10  13.38.55      202
--ERROR--   Line: LINE23 down, auto-recognition failed

****** ALARM FROM MTI_83             85/10/10  13.40.28      202
--ERROR--   Line: LINE23 down, auto-recognition failed
```

DISPLAY_CATENET_TITLES (DISCT)

Purpose Displays the system, community, internal and external titles in the catenet that are registered through the Directory Management Entity (ME).

Format **DISPLAY_CATENET_TITLES**
DISPLAY_OPTION = list of keyword value

Parameters *DISPLAY_OPTION*

Specifies what type (one or more) of titles to display. The following keyword values are allowed.

Keyword Value	Description
SYSTEM (S)	Displays titles of all CDCNET systems (DIs) known in the CDCNET network.
COMMUNITY (C)	Displays titles of DI communities. For this release of CDCNET, the only supported community is all the DIs in the catenet, which has the title CATENET.
INTERNAL_SERVICE (IS)	Displays titles of services that are known only to network services and the command MDI. Internal titles are internal to CDCNET and are not visible to network users.
EXTERNAL_SERVICE (ES)	Displays titles of services that are known to external users. The titles are known only to the command MDI. External titles are available or visible to all the network operators and network users.
ALL	Displays system, community, internal service, and external service titles.

Default is SYSTEM.

Responses Catenet titles (followed by the titles display--see examples). If a specified type is not registered through the Directory ME, the following response is inserted in the display:

None were found.

Remarks For more information on Directory ME, titles, and internal and external services, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
display_catenet_titles

Catenet Titles
 community titles
    CATENET

system titles
  NORTH_ENGIN_BLD_TDI                SOUTH_ENGIN_BLD_TDI
  ENGINEER_CYBER_MDI                 ADMIN_BLD_TDI_1
  ADMIN_BLD_TDI_2                    ADMIN_BLD_NDI_TRUNK
  HDQTRS_BLD_TDI_1                   HDQTRS_BLD_TDI_2
  HDQTRS_CYBER_MDI                   ENG_HDQTRS_NDI_TRUNK

internal_service titles
  ENGINEERING                        HDQTRS

external_service titles
  NP_GW_ENGINEERING                  NP_GW_HDQTRS
```

DISPLAY_COMMAND_INFORMATION (DISCI)

Purpose

Displays the parameters and parameter syntax for a specified session command. This command is identical to DISPLAY_COMMAND_INFORMATION (described in chapter 8) which displays information on network commands. The only difference between the these two commands is that the DISPLAY_COMMAND_INFORMATION command which gives a list of network commands is always embedded within the SEND_COMMAND command.

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 session commands.

ACTIVATE_ALARMS

BYE

CHANGE_ALARM_ENVIRONMENT

DEACTIVATE_ALARMS

DISPLAY_ALARM_ENVIRONMENT

DISPLAY_ALARM_HISTORY

DISPLAY_CATENET_TITLES

DISPLAY_COMMAND_INFORMATION

DISPLAY_COMMAND_LIST

DISPLAY_COMMAND_LIST_ENTRY

DISPLAY_CONNECTED_MDI

EXECUTE_COMMAND_FILE

GOODBYE

HELLO

HELP

INCLUDE_FILE

LOGIN

LOGOUT

QUIT

RESTORE_ALARM_ENVIRONMENT

ROUTE_ALARM

ROUTE_COMMAND_RESPONSE

SEND_COMMAND

SEND_COMMAND_SEQUENCE

SET_COMMAND_MDI

Responses List of parameter names, parameter abbreviations, and parameter syntax for the specified command. (See example.)

--ERROR--Parameter COMMAND is required but was omitted.

--ERROR--The following parameter value, <string> is not a valid command name.

Examples

```
display_command_information command=include_file

file,fname = $required
username, unname = $optional
```

DISPLAY_COMMAND_LIST (DISCL)

Purpose Displays the list of session control commands for which you are validated.

Format **DISPLAY_COMMAND_LIST**

Responses <Alphabetical list of commands legal for network operator's command privilege. See example.>

Examples

```
display_command_list

activate_alarms
bye
change_alarm_environment
deactivate_alarms
display_alarm_environment
display_alarm_history
display_catenet_titles
display_command_information
display_command_list
diplayY_command_list_entry
display_connected_mdi
execute_command_file
goodbye
hello
help
include_file
login
logout
quit
restore_alarm_environment
route_alarm
route_command_response
send_command
send_command_sequence
set_command_mdi
```

DISPLAY_COMMAND_LIST_ENTRY (DISCLE)

Purpose The DISPLAY_COMMAND_LIST_ENTRY command is an alias of the DISPLAY_COMMAND_LIST. Like the DISPLAY_COMMAND_LIST command, this command displays an alphabetical list of the session control commands for which you are validated.

Format **DISPLAY_COMMAND_LIST_ENTRY**

Responses Refer to the description of the DISPLAY_COMMAND_LIST command for further response information.

Examples

```
display_command_list_entry

activate_alarms
bye
change_alarm_environment
deactivate_alarms
display_alarm_environment
display_alarm_history
display_catenet_titles
display_command_information
display_command_list
diplayY_command_list_entry
display_connected_mdi
execute_command_file
goodbye
hello
help
include_file
login
logout
quit
restore_alarm_environment
route_alarm
route_command_response
send_command
send_command_sequence
set_command_mdi
```

DISPLAY_CONNECTED_MDI (DISCM)

Purpose Displays the coupler numbers, system titles, and operational status of the MDIs and MTIs physically connected to the host mainframe.

On NOS, you can only control the network (by sending commands and receiving responses through NETOU) through the MDI which you have selected for communication with the network. This command is executed automatically during the login process. This display contains one line for each CDCNET connection established.

Format **DISPLAY_CONNECTED_MDI**

Parameters None.

Remarks MDIs may have the following operational states.

SELECTED	You are currently communicating with this MDI.
ACTIVE	MDI has been selected previously, and connection has been retained.
AVAILABLE	MDI is available for selection.
UNAVAILABLE	MDI is not available for selection. This state indicates that the MDI has started to establish communication with NETOU but it is not yet ready to allow operator sessions to be established. The system title is not known at this time. If the MDI remains in this state, the MDI is probably hung.

Responses STATUS OF CONNECTED MDIs
(See example.)

No path to CDCNET available.

Examples

```
display_connected_mdi

STATUS OF CONNECTED MDIs
NODE   CURRENT   SYSTEM
NUMBER STATE     TITLE

 3      SELECTED       MTI_83
 5      ACTIVE         MDI_84
 6      AVAILABLE      MDI_8A
 7      UNAVAILABLE    --UNKNOWN--
```

EXECUTE_COMMAND_FILE (EXECF)

Purpose

Directs NETOU to read the named file in your catalog. The INCLUDE_FILE command is an alias for this command. The file may contain any network operations commands, except the EXECUTE_COMMAND_FILE, INCLUDE_FILE, and SET_COMMAND_MDI commands.

For example, you may construct a file that contains commands to select the alarm messages for you. The file's contents are interpreted as one or more commands. These commands may be any operator environment or network commands that address systems or communities within your domain of control.

Format

EXECUTE_COMMAND_FILE
FILE = file name
USER_NAME = name

Parameters

FILE (F)

The name of the file in your catalog or in the catalog of the alternate user name, specified by the USER_NAME parameter (below). The file name follows the NOS rules for file names. The file must be an indirect access permanent file, residing on the default family.

USER_NAME (UN)

Specifies the user name of an alternate catalog in which the command file is located. If the command file is in any other catalog than your own, you must specify this parameter.

Remarks

The following network operations commands cannot be used in a command file: EXECUTE_COMMAND_FILE, INCLUDE_FILE and SET_COMMAND_MDI.

File contents can be in display code if you do not use characters that are not supported in display code, such as ^ and @. If you do use non-display code characters, file contents must use the ASCII 6/12 character code set. Enter the NOS ASCII command prior to creating command files to ensure this; otherwise, use the NOS FCOPY command to change the command file's character code set from another set to the ASCII 6/12 character code set. Reading of the file is terminated at the first end-of-record (EOR) or end-of-file (EOF) encountered.

If you are going to access the command file from another user name with the USER_NAME parameter, the file must be public, semi-private, or private with read access permitted to you.

Secondary user statements executed within IAF have no effect. The default user name reverts back to the original login user name.

You may stop execution of the file by entering a user break 1 or 2 sequence.

If a command inside the command file aborts or causes an error, execution of the command file ceases. For example, if a ROUTE_COMMAND_RESPONSE command in a command file gets a PFM error, execution of the command file ceases.

Responses

Command file <file_name> executing.

(This response is then followed by the responses to the commands in the file (unless responses are routed to a file).

Last command in command file is incomplete and is discarded.

--ERROR--EXECUTE_COMMAND_FILE or INCLUDE_FILE command format error, <file_name> not specified.

--ERROR-- Command file <file_name> not found under username <un>.

--ERROR-- EXECUTE_COMMAND_FILE command is not valid in command file.
Processing of command file is terminated.

--ERROR-- SET_COMMAND_MDI command is not valid in command file.
Processing of command file is terminated.

--ERROR--File <file_name> is a direct access file, should be indirect access.

Examples

The following command directs a file of statistics control commands called NETSTAT to be read and executed:

```
execute_command_file file=netstat

Command file NETSTAT executing

 :
```

(This section contains responses to the commands in NETSTAT.)

HELP

Purpose Performs the same function as the DISPLAY_COMMAND_LIST command. Refer to the DISPLAY_COMMAND_LIST command previously described in this chapter.

Format **HELP**

Responses Refer to the description of the DISPLAY_COMMAND_LIST command for response information.

Examples

```
help

activate_alarms
bye
change_alarm_environment
deactivate_alarms
display_alarm_environment
display_alarm_history
display_catenet_titles
display_command_information
display_command_list
diplayY_command_list_entry
display_connected_mdi
execute_command_file
goodbye
hello
help
include_file
login
logout
quit
restore_alarm_environment
route_alarm
route_command_response
send_command
send_command_sequence
set_command_mdi
```

INCLUDE_FILE (INCF)

Purpose

Performs the same functions as the EXECUTE_COMMAND_FILE command. Refer to the EXECUTE_COMMAND_FILE command previously described in this section.

Format

INCLUDE_FILE
FILE = *name*
USER_NAME = *name*

Parameters

FILE (F)

The name of the file in your catalog or in the catalog of the alternate user name, specified by the USER_NAME parameter (below). The file name follows the NOS rules for file names. The file must be an indirect access permanent file, residing on the default family.

USER_NAME (UN)

Specifies the user name of an alternate catalog in which the include_file is located. If the include_file is in any other catalog than your own, you must specify this parameter.

Remarks

The following network operations commands cannot be used in an INCLUDE_FILE command file: EXECUTE_COMMAND_FILE, SET_COMMAND_MDI, or INCLUDE_FILE.

File contents can be in display code if you do not use characters that are not supported in display code, such as ^ and @. If you do use non-display code characters, file contents must use the ASCII 6/12 character code set. Enter the NOS ASCII command prior to creating command files to ensure this; otherwise, use the NOS FCOPY command to change the command file's character code set from another set to the ASCII 6/12 character code set. Reading of the file terminates at the first end-of-record (EOR) or end-of-file (EOF) encountered.

If you are going to access the command file from another user name with the USER_NAME parameter, you must be permitted to read access to the file.

You may stop execution of the file by entering a user break 1 or 2 sequence.

If a command inside the include_file aborts or causes an error, execution of the include_file ceases. For example, if a ROUTE_COMMAND_RESPONSE command in an include_file gets a PFM error, execution of the include_file ceases.

Responses

Command file <file_name> executing.

(This response is then followed by the responses to the commands in the file (unless responses are routed to a file).

Last command in command file is incomplete and is discarded.

--ERROR--INCLUDE_FILE command format error, <file_name> not specified.

--ERROR-- Command file <file_name> not found under username <user_name>.

--ERROR-- INCLUDE_FILE command is not valid in command file. Processing of command file is terminated.

--ERROR-- SET_COMMAND_MDI command is not valid in command file. Processing of command file is terminated.

--ERROR--File <file_name> is a direct access file, should be indirect access.

Examples

The following command directs a file of statistics control commands called NETSTAT to be read and executed.

```
include file file=netstat

Command file NETSTAT executing

:                                   (This section contains responses to the
                                     commands in NETSTAT.)

File NETSTAT complete
```

QUIT

Purpose Terminates the Network Operator Utility (NETOU) session. Once the QUIT command executes, NETOU commands will not be valid during a NOS command entry session.

Format **QUIT**

Examples

```
NOU/quit
```

Remarks The following session control commands perform the same function and are used in the same way as the QUIT command.

BYE

GOODBYE

HELLO

LOGIN

LOGOUT

RESTORE_ALARM_ENVIRONMENT (RESAE)

Purpose Restores a changed alarm environment to the environment that was defined at operator session login. Reenables receipt of alarms from DIs by a network operator. Reenables disabled alarm communities.

Format **RESTORE_ALARM_ENVIRONMENT**

Responses Alarm environment restored.

Remarks Use this command after your alarm environment has been changed by the CHANGE_ALARM_ENVIRONMENT or DEACTIVATE_ALARMS commands to return to the original set of DIs that report alarms to you.

Examples

```
restore_alarm_environment

Alarm environment restored.
```

ROUTE_ALARM (ROUA)

Purpose

Routes all alarms to a specified direct access file. If you enter this command at the start of your operations session, all alarms that follow will be routed to a file. At the start of an operations session, routing to the operations console display is assumed.

If a direct access file to receive alarms already exists, subsequent alarms are appended to the end of the file. If the file does not exist, NETOU defines the file. If the file is busy, or the named file is an indirect access permanent file, the command will fail. Both command responses and alarms may be routed to the same file.

Format

ROUTE_ALARM
FILE = list 1..2 of name and/or keyword value

Parameters

FILE (F)

The name of the file to receive the alarms. The alarm file is a text file that uses the ASCII 6/12 character code set. This file must be a NOS direct access permanent file. You may define the same file to receive both alarms and command responses. If the file does not exist, NETOU will define the file. If the file already exists, NETOU appends subsequent alarms to the end of the existing file.

The keyword value DISPLAY indicates that you want alarms to be returned to your terminal or console screen. Both a file name and DISPLAY may be entered as a list in parentheses. In that case, alarms are both recorded in the file and returned to your display. If you specify a file name and DISPLAY, only one file name may be specified. The file name follows the NOS rules for file names. If you specify no file name, the default DISPLAY is assumed, and alarms will be returned to your terminal or console screen.

Responses

Alarms routed to <file_name>.

Alarms routed to DISPLAY and <file_name>.

--ERROR-- Illegal ROUTE command, when more than one file is specified, one must be DISPLAY.

--ERROR-- Illegal ROUTE command, DISPLAY specified more than once.

--ERROR-- File <file_name> is an indirect access file, should be direct access.

Remarks

Entering a second ROUTE_ALARM command terminates the alarm routing from a previous command.

Examples

This example shows the ROUTE_ALARM command establishing that alarms are to be routed to both a direct access file called TODLOG and to the operations station display.

```
route_alarm file=(todlog,display)

Alarms routed to DISPLAY and TODLOG.
```

ROUTE_COMMAND_RESPONSE (ROUCR)

Purpose Routes all command responses to a specified direct access file. This command is used only in NOS-based CDCNET operations. If you enter this command at the start of your operations session, or make it part of your operator's user prolog, all command responses that follow will be routed to a file. This command allows you to review lengthy responses, such as status and configuration displays, using a listing of the responses.

Format **ROUTE_RESPONSE**
FILE = list 1..2 of name or keyword value

Parameters *FILE (F)*

The name of the file to receive the responses. This file must be a NOS direct access permanent file. Note that both command responses and alarms may be routed to the same file. A command response file is a NOS direct access text file that uses the ASCII 6/12 character code set.

If the file does not exist NETOU will define the file. If the file already exists, NETOU appends subsequent command responses to the end of the file. If the file name specified is DISPLAY, responses are routed to your operations station. At the start of an operations session, routing to DISPLAY is assumed. If you specify a file name and DISPLAY, only one file name may be specified. If you specify no file name, the default DISPLAY is assumed.

Responses Command responses routed to <file_name>.

Command responses routed to DISPLAY and <file_name>.

--ERROR-- Illegal ROUTE command, when more than one file is specified, one must be DISPLAY.

--ERROR-- Illegal ROUTE command, DISPLAY specified more than once.

--ERROR-- File <file_name> is an indirect access file, should be direct access.

Remarks Entering a second ROUTE_COMMAND_RESPONSE command terminates the command response routing from a previous command.

Examples This example shows the ROUTE_COMMAND_RESPONSE command establishing that command responses are to be routed to both direct access file TODLOG and to the operations station display. This example and the ROUTE_ALARM command example show messages being routed to the same file (TODLOG).

```
route_response file=(todlog,display)

Command responses routed to DISPLAY and TODLOG.
```

SEND_COMMAND (SENC) (NOS Version)

Purpose

Sends the CDCNET command to a DI or list of DIs. All NETOU commands except session control commands must be enclosed within SEND_COMMAND in order for the commands to get to the appropriate DIs. The maximum size of the total SEND_COMMAND string is 512 characters. The maximum size of the command string within SEND_COMMAND is 256 characters.

Format

SEND_COMMAND
COMMAND = string
SYSTEM = list 1..15 of name

Parameters

COMMAND (C)

The network operations command to be sent to the specified DI. Enter the command as a string value enclosed by apostrophes ('). You may use the abbreviated form of the command. If the command you are sending contains a string value (such as WRITE_TERMINAL_MESSAGE), you must use two consecutive apostrophes at the beginning and end of the string in order for the enclosed string to be recognized (see examples). You cannot substitute the quotation mark character for two apostrophes.

SYSTEM (S)

The logical or physical DI name or list of DI names to which the command is to be sent. If you omit this parameter, the name of the last DI or list of DIs to which you sent a command is used. The default DI for the first use of SEND_COMMAND during your session is the MDI through which you are connected to the network.

Remarks

You do not have to use SEND_COMMAND for session control commands. Refer to the Session Control chapter for the complete list of session control commands for NOS environments.

Examples

The following command sends a DISPLAY_DI_SYSTEM_STATUS command to the DI named North_TDI_1.

```
send_command command='display_di_system_status' ..
system=north_tdi_1
```

The following command sends a DISPLAY_DI_SYSTEM_STATUS command to DIs North_TDI_1 and East_TDI_2.

```
send_command command='display_di_system_status'..
system=(north_tdi_1,east_TDI_2)
```

This SEND_COMMAND example shows how to send a command that also contains a string value. In this case, the command is WRITE_TERMINAL_MESSAGE, and it is being sent to terminal users connected to TDI1. The WRITE_TERMINAL_MESSAGE command is surrounded by apostrophes, and the string value within the command (the message to the terminal users) is designated by two consecutive apostrophes.

```
send_command c='write_terminal_message m=''New communications ..
configuration tomorrow''',system=tdi1
```

SEND_COMMAND_SEQUENCE (SENCS)

Purpose

Allows you to send one or more commands to the same system(s) without enveloping the command within a SENC command. This command puts you in a different mode (SENCS mode). After entering the SENCS command, you receive a SENCS/ prompt. All commands you enter following the SENCS/ prompt will be sent only to those systems specified in the system parameter of the command. As you enter each command, the command is sent to the specified system(s) for processing.

The SENCS command may be included in a prolog or command file. If so, all subsequent commands will be sent directly to the system specified by the command.

To leave the SENCS mode, you enter **. If a prolog or command file contains the SENCS command, all subsequent commands on that file are sent to the specified system for processing until a ** is detected.

If you wish to send a network command to other systems while in the SENCS mode an escape character, a single asterisk '*', is provided. To use this you type the escape character, '*' followed by the network command on the same line. This one command then will be sent to the specified systems and need not be encapsulated within the SENC command. Subsequent commands are again processed in the SENCS mode unless they are preceded with the escape character. When a command is continued on more than one line, the '*' applies only to the first line. In other words, if '*' is entered anywhere in the subsequent lines, it will be treated as part of the command text.

Format

SEND_COMMAND_SEQUENCE
SYSTEM = list 1..15 of system titles

Parameters

SYSTEM (S)

The logical or physical DI name or list of DI names to which the command is to be sent. A maximum of 15 system titles may be specified.

Responses

Entering SENCS mode, type ** to exit.

SENCS/ --ERROR-- Parameter SYSTEM is required but was omitted.

Examples

The following command sends a DISPLAY_DI_SYSTEM_STATUS command to the DI named North_TDI_1.

```
send_command_sequence system = north_tdi_1
SENCS/display_di_system_status
SENCS/**
```

The following command sends a DISPLAY_DI_SYSTEM_STATUS command to DIs North_TDI_1 and East_TDI_2.

```
send_command_sequence system=(north_tdi_1,east_tdi_2)
SENCS/display_di_system_status
SENCS/**
```

SET_COMMAND_MDI (SETCM)

Purpose

Selects the MDI (or MTI) through which you send commands to the network and from which you receive responses and alarms from the network. At any time, you can communicate with only one MDI. If only one DI (MDI or MTI) is connected to a host, this command is not needed. It is only needed in configurations supporting more than one MDI or MTI per host. When you select an MDI for the first time, your user prolog automatically executes. Subsequent, consecutive selection of the same MDI causes recovery of the operator environment for that MDI. Using this command, you may switch communications from one MDI to another. Whenever you select a different MDI, the session with the currently selected MDI is broken. You may specify whether the operations session should be terminated with the old MDI (using the RETAIN parameter).

You will receive responses only from the currently selected MDI. However, you will receive alarms from all MDIs with which you have active NETOU sessions. If a session with a previously selected MDI is retained (see RETAIN parameter) and the previously selected and currently selected MDI are in the same catenet, it is possible that you will receive the same alarm twice; once from each MDI. Because of this, the RETAIN parameter should only be used when switching between MDIs belonging to disjoint catenets.

Format

SET_COMMAND_MDI
MDI = name
RETAIN = boolean

Parameters

MDI (M)

The system name of the MDI or MTI to which your operations session is switched. If you omit this parameter, NETOU attempts to use the MDI specified on the NETOU job statement as the default MDI to be selected, if an MDI is specified and it is available. Otherwise, NETOU will select the longest-connected available MDI as the default.

RETAIN (R)

Indicates whether or not the operations session with the currently selected MDI or MTI should be retained. Possible values are YES, Y, NO or N. Default is NO. If you select YES, the current session is retained. You may subsequently resume that session using another SET_COMMAND_MDI command. If you select NO, the operations session with the MDI or MTI you have been using is ended. If you are switching between MDIs or MTIs on disjoint catenets, RETAIN should be set to YES. NETOU will display received alarms for both the retained session and your currently selected session at your operations session as well as sending them to the alarm history buffer. You may also review the alarms for a retained session using the DISPLAY_ALARM_HISTORY command. If you are switching between MDIs or MTIs on a common catenet, RETAIN should be set to NO. This prevents NETOU from displaying duplicate alarms for the new and previous sessions.

Responses MDI selected = <system_title>

--ERROR-- The value <value> is not valid as a RETAIN option.

--ERROR-- MDI not available, MDI = <system_name>.

Remarks This command cannot be contained in a CDCNET network operations command file.

Examples

```
set_command_mdi mdi=mdi_3

MDI selected = MDI_3
```

** Command

Purpose Terminates the SEND_COMMAND_SEQUENCE execution mode.

Format **

Remarks Use this command when in SENCS mode. The command allows you to exit (quit) the SENCS mode of execution begun when you entered the SENCS command earlier in your session.

Examples

```
SENCS/**
```

Network Commands 8

Network Commands 8

This chapter provides complete descriptions of all CDCNET Network Operations commands. Command descriptions are grouped by type.

The format for the command descriptions in this chapter is located on the divider page for part 2 of this manual.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

TCP/IP commands are used to monitor and control an environment implementing TCP/IP protocols through CDCNET. Use these commands when your operations environment includes not only NOS/VE but also foreign hosts through the TCP/IP network.

NOTE

To get commands to the proper DI(s), embed each command within SEND_COMMAND: SEND_COMMAND COMMAND='command',SYSTEM=name. Each command description shows the use of the SEND_COMMAND to ensure command delivery to the proper DI(s).

To get a series of commands to the same DI without repeatedly entering the SENC command, enter the SEND_COMMAND_SEQUENCE (SENCS) command. The SENCS command puts you in SENCS mode, which allows you to send multiple commands to a single DI without entering multiple SENC commands. SENCS mode is described in chapters 6 and 7 (NOS/VE and NOS, respectively).

CANCEL Commands

CANCEL_CHANNEL_NET (CANCN)

Purpose Cancels the configuration of a channel network and the underlying channel trunk definition. The network must have been previously stopped. Address the network by its logical name.

Format

CANCEL_CHANNEL_NET
NETWORK_NAME = name

Parameters

NETWORK_NAME (NN)

The logical name of the network, assigned by the DEFINE_CHANNEL_NET command that configured the network.

Responses

CHANNEL network <network_name> cancelled for trunk (trunk_name).

--WARNING-- Network <network_name> is not defined.

--ERROR-- Network <network_name> is active. It must be stopped before being cancelled.

--ERROR-- Network <network_name> is not an CHANNEL network.

Examples

```
senc c='cancel_channel_net network_name = cyber101_log_link'

CHANNEL network CYBER_101_LOG_LINK cancelled for trunk
CYBER_101_COUPLER_3.
```

CANCEL_CHANNEL_TRUNK (CANCT)

Purpose Cancels the configuration of a channel trunk. Address the trunk by its logical name.

Format **CANCEL_CHANNEL_TRUNK**
TRUNK_NAME = name

Parameters **TRUNK_NAME (NN)**

The logical name of the trunk, assigned by the DEFINE_CHANNEL_TRUNK command that configured the trunk.

Responses CHANNEL <trunk_name> cancelled.

--WARNING-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> active, cannot be cancelled.

--ERROR-- Trunk <trunk_name> is not a CHANNEL trunk.

--ERROR-- Channel Trunk <trunk_name> cannot be cancelled until NP Interface <Interface_name) is cancelled.

--ERROR-- Channel Trunk <trunk_name> cannot be cancelled until Channel Network <network_name> is cancelled.

Examples

```
senc c='cancel_channel_trunk trunk_name = cyber101_alt'

CHANNEL trunk CYBER_101_ALT cancelled.
```

CANCEL_DEVICE_OUTCALL_SERVICE (CANDOS)

Purpose Cancels the definition of the Device Outcall Service. Because a DI supports only one device outcall at a time, the command requires no parameter to identify the service being cancelled.

Format **CANCEL_DEVICE_OUTCALL_SERVICE**

Responses Device Outcall Service cancelled.

--WARNING-- Device Outcall Service not defined.

Examples

```
senc c='cancel_device_outcall_service'

Device Outcall Service cancelled.
```

CANCEL_ETHER_NET (CANEN)

Purpose

Cancels the configuration of an Ethernet network and the underlying Ethernet trunk. The network is addressed by its logical name. The network must be stopped by the STOP_NETWORK command before it can be cancelled.

Format

CANCEL_ETHER_NET
NETWORK_NAME = name

Parameters

NETWORK_NAME (NN)

The logical name of the network, assigned by the DEFINE_ETHER_NET command that configured the network.

Responses

Ethernet network <network_name> cancelled for trunk <trunk_name>.

--WARNING-- Network <network_name> not defined.

--ERROR-- Network <network_name> is active. It must be stopped before being cancelled.

--ERROR-- Network <network_name> is not an Ethernet network.

Examples

```
senc c='cancel_ether_net network_name=engin_bldg_net',s=engin_ndi_1

Ethernet network ENGIN_BLDG_NET cancelled for trunk ENGIN_BLDG_TRUNK.
```

CANCEL_ETHER_TRUNK (CANET)

Purpose

Cancels the configuration of an Ethernet trunk. The trunk is addressed by its logical name. This command also cancels an Ethernet network, if one exists at the time this command is issued. Ethernet trunk cancellation can also be done by the CANCEL_ETHER_NET command, which cancels an Ethernet network and its underlying trunk.

Format

CANCEL_ETHER_TRUNK
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

The logical name of the trunk, assigned by the DEFINE_ETHER_TRUNK command that configured the trunk.

Responses

Ethernet trunk <trunk_name> cancelled.

--WARNING-- Trunk <trunk_name> not defined.

--ERROR-- Network <network_name> is active. It must be stopped before being cancelled.

--ERROR-- Trunk <trunk_name> is not an Ethernet trunk.

Examples

```
senc c='cancel_ether_trunk trunk_name=engin_bldg_nsouth',s=engin_ndi_1

Ethernet trunk ENGIN_BLDG_SOUTH cancelled.
```

CANCEL_FILE_SUPPORT (CANFS) (NOS Only)

Purpose

Cancels support for access of the specified file types from an MDI or MTI connection to a NOS host. If all file types defined for the host are cancelled, file access from the NOS host is terminated.

Format

CANCEL_FILE_SUPPORT
FILE_TYPE = list 1..8 of keyword value
TRUNK_NAME = name

Parameters

FILE_TYPE (FT)

A list of one or more file types to be cancelled. The following file types are allowed.

EXCEPTION
BOOT
DUMP
LIBRARY
CONFIGURATION
TERMINAL_PROCEDURE
USER_PROCEDURE
LOAD_PROCEDURE
ALL

Default is ALL.

TRUNK_NAME (TN)

The trunk name of the logical link to the host for which support of the specified file types is cancelled. The value for this parameter is determined by the DI load process. Its use is not recommended. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is either the trunk name as specified by a DEFINE_SYSTEM command or the trunk over which the DI was loaded.

Responses

File Support for specified file_types is cancelled for trunk <name>.

--WARNING-- File Support for specified file_types is cancelled for trunk <name>. <name> file_type was not defined.

--WARNING-- File Support was not defined for trunk <name>.

--WARNING-- File Support was not defined for the system.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

Examples

```
senc c='cancel_file_support file_type=dump',s=mdi2

File Support for specified file_types is cancelled for trunk 02.
```

CANCEL_HDLC_NET (CANHN)

Purpose Cancels the configuration of an HDLC network and the underlying HDLC trunk definition. The network is addressed by its logical name.

Format **CANCEL_HDLC_NET**
NETWORK_NAME = name

Parameters **NETWORK_NAME (NN)**

The logical name of the network assigned by a DEFINE_HDLC_NET command.

Responses HDLC network <network_name> cancelled for trunk <trunk_name>.

--WARNING-- Network <network_name> is not defined.

--ERROR-- Network <network_name> is active. It must be stopped before being cancelled.

--ERROR-- Network <network_name> is not an HDLC network.

Examples

```
senc c='cancel_hdlc_net network_name = menlo_park_network'

HDLC network MENLO_PARK_NETWORK cancelled for trunk MENLO_PARK_TRUNK.
```

CANCEL_HDLC_TRUNK (CANHT)

Purpose Cancels the configuration of an HDLC trunk. The trunk is addressed by its logical name. The HDLC trunk cannot be cancelled unless the network has been previously cancelled.

Format

CANCEL_HDLC_TRUNK
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

The logical name of the trunk, assigned by the DEFINE_HDLC_TRUNK command that configured the trunk.

Responses

HDLC trunk <name> cancelled.

--WARNING-- Trunk <name> not defined.

--ERROR-- Trunk <name> is not an HDLC trunk.

--ERROR-- A network must be cancelled for trunk <name> before it can be cancelled.

Examples

```
senc c='cancel_hdlc_trunk trunk_name = menlo_park_trunk_1'

HDLC trunk MENLO_PARK_TRUNK_1 cancelled.
```

CANCEL_IP_HOST (CANIH)

Purpose

Cancels the definition of an Internet Protocol (IP) host and its associated routing information. The host is referenced by its IP network address.

Format

CANCEL_IP_HOST
IP_ADDRESS = list 4 of 0..255

Parameters

IP_ADDRESS (IA)

The IP address of the host, defined by the DEFINE_IP_HOST command that configured the host. The format is similiar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

Responses

IP Host address <ip_address> is canceled.

--WARNING-- IP Host address <ip_address> is not defined.

--ERROR-- IP Host address <ip_address> is invalid.

Examples

```
senc c='cancel_ip_host ip_address=(128,2,53,7)',s=ndi1

IP address 128.2.53.7 is canceled.
```

CANCEL_IP_NET (CANIN)

Purpose

Cancels the definition of an Internet Protocol (IP) network and its associated routing information. The network is referenced by its IP network address.

Format

CANCEL_IP_NET
IP_NETWORK = list 4 of 0..255

Parameters

IP_NETWORK (IN)

The IP address of the network, assigned by a DEFINE_IP_NET command that configured the network. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP network 128.2.0.0 is represented as (128,2,0,0) or (128,2).

Responses

IP Network <ip_network> is canceled.

--WARNING-- IP Network <ip_network> is not defined.

--ERROR-- IP Network <ip_network> is invalid. Only the network number part of an IP address should be specified.

Examples

```
senc c='cancel_ip_net ip_network=(128,2,0,0)',s=mdi1

IP network 128.2.0.0 is canceled.
```

CANCEL_LINE (CANL)

Purpose
Cancels the configuration of a communication line or a URI line. The line is addressed by its logical name.

Format
CANCEL_LINE
LINE_NAME = name

Parameters
LINE_NAME (LN)

The logical name of the line, assigned during configuration by the DEFINE_LINE command.

Responses
Line <line_name> cancelled.

--WARNING-- Line <line_name> not defined.

--ERROR-- Line <line_name> active, cannot be cancelled.

Remarks
The line must first be stopped using the STOP_LINE command before it can be cancelled.

Examples
```
senc c='engin_tdi,c='cancel_line line_name=engin_line_1'

Line ENGIN_LINE_1 cancelled.
```

CANCEL_NP_INTERFACE (CANNI) (NOS Only)

Purpose Cancels the configuration of a NOS Network Products (NP) interface.

Format

CANCEL_NP_INTERFACE
INTERFACE_NAME = name

Parameters

INTERFACE_NAME (IN)

The logical name of the interface assigned by a DEFINE_NP_INTERFACE command.

Responses

NP interface <interface_name> cancelled for trunk <trunk_name>.

--WARNING-- NP interface <interface_name> is not defined.

--ERROR-- NP interface <interface_name> is active. It must be stopped before being cancelled.

--ERROR-- NP interface <interface_name> has active users. They must be cancelled before the NP interface can be cancelled.

Examples

```
senc c='cancel_np_interface in=cyber_109',s=mdi1

NP interface CYBER_109 cancelled for trunk $MCI2.
```

CANCEL_OPERATOR_SUPPORT (CANOS) (NOS Only)

Purpose

Cancels support for the Operator Support Application in an MDI or MTI. The Operator Support Application allows network operators to communicate with the network DIs through a particular MDI or MTI, using the Network Operator Utility (NETOU).

This command suppresses all responses to outstanding commands from operators connected through the host.

Format

CANCEL_OPERATOR_SUPPORT
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

The trunk name of the logical link to the host for which operator support is cancelled. If this parameter is not specified, the default value is used.

Responses

Operator Support is cancelled for trunk <name>.

--WARNING-- Operator Support was not defined for trunk <name>.

--WARNING-- Operator Support was not defined for the system.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--FATAL-- Unable to cancel Operator Support for trunk <name>.

Examples

```
senc c='cancel_operator_support trunk_name=c170_trunk1'

Operator Support is cancelled for c170_trunk
```

CANCEL_PASSTHROUGH_SERVICE (CANPS)

Purpose Cancels the definition of the passthrough service currently supported by a DI. Because a DI supports only one passthrough service at a time, the command requires no parameter to identify the passthrough service being cancelled.

Format **CANCEL_PASSTHROUGH_SERVICE**

Parameters None.

Responses Passthrough service cancelled.

--WARNING-- Passthrough Service not defined.

Examples

```
senc c='cancel_passthrough_service'

Passthrough service cancelled.
```

CANCEL_RECORDER_LOG_GROUP (CANRLG) (NOS Only)

Purpose

Cancels the recording of a log group at an MDI or MTI connected to a NOS host. The Independent Log Management Entity in the MDI will no longer record the cancelled group (the recorder DI will no longer log the source log messages to the network log file on the host). If no groups are recorded by an Independent Log ME, this command terminates the log recording function. If no other recorders are defined for the log group, this command always terminates the Independent Log ME log recording function for the DI and the log group. The current CDCNET release supports only one log group per MDI or MTI.

Format

CANCEL_RECORDER_LOG_GROUP
LOG_GROUP = name
TRUNK_NAME = name

Parameters

LOG_GROUP (LG)

Specifies the logical name of the log group for which support is to be cancelled. The default log group supported for this release is CATENET. A DI can belong to only one log group. Each recorder DI supports one named log group.

TRUNK_NAME (TN)

The trunk name of the logical link to the host for which the specified log groups are cancelled. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is the default channel trunk, as specified by the DEFINE_SYSTEM command or if not specified on the DEFINE_SYSTEM command, the channel trunk over which the DI was loaded.

Responses

Recorder log group is cancelled for trunk <name>.

--WARNING-- Specified recorder log group is cancelled for trunk <name>. Recorder log group <name> was not defined.

--WARNING-- Recorder log groups were not defined for trunk <name>.

--WARNING-- Recorder log groups were not defined for the system.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

Examples

```
senc c='cancel_recorder_log_group lg=catenet tn=c170_trunk1',s=mdi3

Recorder log groups are cancelled for trunk c170_trunk.
```

CANCEL_REMOTE_LOAD_SUPPORT (CANRLS)

Purpose Cancels the support of the CDCNET system for the loading and dumping of systems through the networks directly connected to the CDCNET system. The command stops and deletes the Initialization Management Entity from the system.

Format **CANCEL_REMOTE_LOAD_SUPPORT**

Responses Remote Load Support is cancelled.

-- WARNING -- Remote Load Support is not defined for this system.

Remarks For more information on the Initialization Management Entity, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces. Also see the DEFINE_REMOTE_LOAD_SUPPORT command later in this chapter.

Examples

```
senc c='cancel_remote_load_support'

Remote Load Support is cancelled.
```

CANCEL_SERVER_TELNET_GW (CANSTG)

Purpose Cancels the definition of a server TELNET gateway. The gateway must be stopped using the STOP_SERVER_TELNET_GW command before it can be canceled with this command.

Format **CANCEL_SERVER_TELNET_GW**
GATEWAY_NAME = name

Parameters **GATEWAY_NAME (GN)**

The logical name of the server TELNET gateway defined by a DEFINE_SERVER_TELNET_GW command.

Responses Server TELNET gateway <gateway_name> is canceled.

--WARNING-- Server TELNET gateway <gateway_name> is not defined.

--ERROR-- Server TELNET gateway <gateway_name> is active. It must be stopped before being canceled.

Examples

```
senc c='cancel_server_telnet_gw gateway_name=gw_to_cyber',s=mdi1

Server TELNET gateway GW_TO_CYBER is canceled.
```

CANCEL_SOURCE_ALARM_MESSAGE (CANSAM)

Purpose Cancels the reporting of specified alarm messages by a DI. The message numbers specified are removed from the list of alarms to be sent from a DI.

Format **CANCEL_SOURCE_ALARM_MESSAGE**
MESSAGE_NUMBER = list 1..63 range of 1..32999

Parameters **MESSAGE_NUMBER (MN)**

Specifies alarm message numbers of one or more alarm messages to be cancelled. Refer to the CDCNET Diagnostic Messages manual for the complete list of alarm messages and their identifier numbers.

Responses Source alarm messages cancelled.

Examples

```
senc c='cancel_source_alarm_message mn=(3,42..45,87)',s=tdi3

Source alarm messages cancelled.
```

CANCEL_SOURCE_LOG_GROUP (CANSLG)

Purpose

Cancels the current definition of the logging function for DIs acting as sources of log messages. This release allows definition of only one log group per system; therefore this command cancels all logging by a DI. To reenable logging, a DEFINE_SOURCE_LOG_GROUP command should immediately follow a CANCEL_SOURCE_LOG_GROUP command.

Format

CANCEL_SOURCE_LOG_GROUP
LOG_GROUP = list 1..1 of name

Parameters

LOG_GROUP (LG)

The logical name for the log group to cancel from reporting. The default log group is CATENET.

Responses

Source log group cancelled.

--WARNING-- Specified source log group cancelled. Source log group <name> was not defined.

--WARNING-- No source log groups defined.

Examples

```
senc c='cancel_source_log_group log_group=catenet',s=engin_tdi_3

Source log group cancelled.
```

CANCEL_TCPIP_GW (CANTG)

Purpose

Cancels the definition of an application interface gateway to Defense Data Network (DDN). The gateway must be stopped using the STOP_TCPIP_GW command before it is canceled with this command.

Format

CANCEL_TCPIP_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the TCP/IP application gateway defined by a DEFINE_TCPIP_GW command.

Responses

TCP/IP gateway <gateway_name> is canceled.

--WARNING-- TCP/IP gateway <gateway_name> is not defined.

--ERROR-- TCP/IP gateway <gateway_name> is active. It must be stopped before being canceled.

Examples

```
senc c='cancel_tcpip_gateway gateway_name=ftp_gw',s=mdi1

TCP/IP gateway FTP_GW is canceled.
```

CANCEL_USER_TELNET_GW (CANUTG)

Purpose Cancels the definition of a user TELNET gateway. The gateway must be stopped using the STOP_USER_TELNET_GW command before it can be canceled.

Format

CANCEL_USER_TELNET_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the user TELNET gateway defined by a DEFINE_USER_TELNET_GW command.

Responses

User TELNET gateway <gateway_name> is canceled.

--WARNING-- User TELNET gateway <gateway_name> is not defined.

--ERROR--User TELNET gateway <gateway_name> is active. It must be stopped before being canceled.

Examples

```
senc c='cancel_user_telnet_gw gateway_name=gw_to_vax',s=mdi1

User TELNET gateway GW_TO_VAX is canceled.
```

CANCEL_X25_ASYNCTIP (CANXA)

Purpose Cancels the X.25 asynchronous TIP service supported by the specified X.25 trunk(s).

Format **CANCEL_X25_ASYNCTIP (CANXA)**
TRUNK_NAME = list 1..32 of name

Parameters **TRUNK_NAME (TN)**

The logical name for the trunk(s) for which X.25 asynchronous TIP service is to be cancelled. The logical name for the trunk(s) was assigned by the DEFINE_X.25_TRUNK command that configured the trunk(s).

Responses AsyncTip support cancelled for specified trunks.

--ERROR-- Trunk <trunk_name> is not a X.25 trunk.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- X.25 AsyncTip support not defined for trunk <trunk_name>.

--ERROR-- X.25 AsyncTip support active for trunk <trunk_name>. Service for this trunk must be stopped before being cancelled.

--ERROR--X.25 AsyncTip support active for one or more of the specified trunks. Service for these trunks must be stopped before being cancelled.

Examples

```
senc c='cancel_x25_asynctip trunk_name = telenet_2'

X.25 AsyncTip support cancelled for specified trunks.
```

CANCEL_X25_GW (CANXG)

Purpose Cancels the configuration of an X.25 gateway and the X.25 outcall titles associated with the gateway.

Format **CANCEL_X25_GW**
GATEWAY_NAME = name

Parameters **GATEWAY_NAME (GN)**

The logical name of the gateway assigned by a DEFINE_X25_GW command.

Responses X.25 gateway <gateway_name> is cancelled.

--WARNING-- X.25 gateway <gateway_name> is not defined.

--ERROR-- X.25 gateway <gateway_name> is active. It must be stopped before being cancelled.

Remarks The X.25 gateway must be stopped by a STOP_X25_GW command before it can be cancelled.

Examples

```
senc c='cancel_x25_gw gn=telenet_gw',s=xndi1

X.25 gateway TELENET_GW is cancelled.
```

CANCEL_X25_INTERFACE (CANXI)

Purpose Cancels the configuration of an X.25 interface.

Format **CANCEL_X25_INTERFACE**
INTERFACE_NAME = name

Parameters **INTERFACE_NAME (IN)**

The logical name of the interface assigned by a DEFINE_X25_INTERFACE command.

Responses X.25 interface <interface_name> cancelled for trunk <trunk_name>.

--WARNING-- X.25 interface <interface_name> is not defined.

--ERROR-- X.25 interface <interface_name> is active. It must be stopped before being cancelled.

--ERROR-- X.25 interface <interface_name> has active users. They must be cancelled before the X.25 interface can be cancelled.

Remarks Before the X.25 interface can be cancelled, the interface must be stopped by the STOP_X25_INTERFACE command, and the X.25 gateway and X.25 network must be cancelled by the CANCEL_X25_GW and CANCEL_X25_NET commands.

Examples

```
senc c='cancel_x25_interface in=telenet_2',s=xndi2

X.25 interface TELENET_2 cancelled for trunk TELENET2.
```

CANCEL_X25_NET (CANXN)

Purpose Cancels the configuration of a X.25 network. The network is addressed by its logical name.

Format **CANCEL_X25_NET**
NETWORK_NAME = name

Parameters **NETWORK_NAME (NN)**

The logical name of the network assigned by the define command (DEFINE_X25_NET) that configured the network.

Responses X.25 network <network_name> cancelled for trunk <trunk_name>.

--WARNING-- Network <network_name> is not defined.

--ERROR-- Network <network_name> is active. It must be stopped before being cancelled.

--ERROR-- Network <network_name> is not an X.25 network.

Remarks The X.25 network must be stopped by the STOP_NETWORK command before it can be cancelled.

Examples

```
senc c='cancel_x25_net network_name=tymnet_net_1',s=xndi2

X.25 network TYMNET_NET_1 cancelled for trunk
TYMNET_TRUNK_3.
```

CANCEL_X25_TRUNK (CANXT)

Purpose Cancels the configuration of an X.25 trunk. The trunk is addressed by its logical name.

Format **CANCEL_X25_TRUNK**
TRUNK_NAME = name

Parameters **TRUNK_NAME (TN)**

The logical name of the trunk assigned by the define command (DEFINE_X25_TRUNK) that configured the trunk.

Responses X.25 trunk <name> cancelled.

--WARNING-- Trunk <name> not defined.

--ERROR-- Trunk <name> active, cannot be cancelled.

--ERROR-- Trunk <name> is not an X.25 trunk.

Remarks The X.25 interface for the trunk must be stopped by a STOP_X25_INTERFACE command before the trunk can be cancelled.

Examples

```
senc c='cancel_x25_trunk trunk_name=tymnet_trunk_1',s=xndi2

X.25 trunk TYMNET_TRUNK_1 cancelled.
```

CHANGE Commands

CHANGE_ELEMENT_STATE (CHAES)

Purpose

Changes the operational state of DI hardware. DI hardware may be placed in the OFF, ON, or DOWN state (states described below). If you use this command, you must also stop the communications traffic or the diagnostics being run on the device whose state you are changing, using the appropriate STOP command. DI hardware devices are addressed by their physical names.

Format

CHANGE_ELEMENT_STATE
DEVICE_NAME = name
STATE = keyword value

Parameters

DEVICE_NAME (DN)

The physical name of the device. This name may have the following values.

For boards:	$, board type (0..7) and board slot number, as in $ESCI6.
For LIM ports:	$, the keyword LIM followed by the LIM board slot number and the keyword PORT followed by the port number on the LIM, as in $LIM5_PORT1.

STATE (S)

The desired new state for the device. The following keyword values are allowed.

Keyword Value	Description
OFF	Sets the device as inactive, so that the device cannot be used or have commands sent to it. The only action allowed against a device in the OFF state is to send a CHANGE_ELEMENT_STATE command to change the state from OFF to another state.
ON	Sets the device in the ON state. ON is the required state for using the device for CDCNET communications.
DOWN	Sets the device as available for diagnostics only. Executing a diagnostic test for a device changes its state to DOWN. If the diagnostic fails, the device remains in the DOWN state. If the diagnostic test passes, the device is placed in the ON state.

Responses

Device <device_name> <state> (ON, OFF, or DOWN).

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> active, stop communications or diagnostics before changing device state.

Examples

```
senc c='change_element_state device_name=$cim4,state=down',s=tdi5

Device $CIM4 down.
```

CHANGE_PASSTHROUGH_SERVICE (CHAPS)

Purpose

Changes the passthrough default inactivity timer.

Format

CHANGE_PASSTHROUGH_SERVICE
INACTIVITY_TIMER = keyword value

Parameters

INACTIVITY_TIMER (IT)

The inactivity timer measures the time period during which no data is being sent in either direction over a paired passthrough connection. This parameter specifies the maximum time, in seconds, that a passthrough connection can be idle. When the time period specified by this parameter (or the default value) expires, the passthrough connection to the terminal user disconnects. The newly-selected timer value affects new as well as existing connections that use the default timer. Specify the timer value in seconds (range of 120 .. 14400). The default value is INFINITE.

Responses

Change of Passthrough Service accepted.

--ERROR-- Passthrough Service not defined.

Examples

```
senc c='change_passthrough_service it=30'

Change of Passthrough Service accepted.
```

CHANGE_SERVER_TELNET_GW (CHASTG)

Purpose Changes the operational parameters of a server TELNET gateway. The original values for these parameters were specified (or defined as defaults) in the DEFINE_SERVER_TELNET_GW command. Any changes specified affect only new connections to the gateway; existing connections are not affected.

Format

CHANGE_SERVER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = name
MAX_CONNECTIONS = 0..65535 or *keyword value*
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..2147483647
TCP_TIMEOUT = 0..65535 or *keyword value*
INACTIVITY_TIMEOUT = 0..65535

Parameters

GATEWAY_NAME (GN)

The logical name of the server TELNET gateway used in subsequent commands that reference the gateway.

IP_ADDRESS (IA)

The IP address of the host for which this gateway provides server TELNET terminal service. The format is similar to the decimal octet convention used by the TCP/IP community, except that the periods are replaced with commas, and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

TITLE or *TITLES (T)*

Specifies the title that this gateway translates to locate the service provider. If the destination system is NOS, this title must be from the DEFINE_NP_TERMINAL_GW command. If the destination system is NOS/VE, this title must be the one registered by the terminal manager.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the CDCNET catenet that should be searched for the service corresponding to the title information given in the TITLE parameter. The only supported value is CATENET.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous connections to be supported by the gateway. If INFINITE is entered, there is no restriction to the number of connections allowed.

TCP_PORT_NUMBER (TPN)

Specifies the TCP port number to be used by the gateway. The default is the well-known server TELNET port 23. Server TELNET issues a TCP PASSIVE_CONNECT request using the well-known port for the source port.

TCP_ALLOCATE_SIZE (TAS)

Specifies the amount of data that the gateway queues for each connection. Larger values might improve user response time, especially for PC users (with a standard protocol such as XMODEM), but might also increase the number of instances of DI congestion.

CAUTION

Changing this value is discouraged, and should be done with caution, as network service may be disrupted.

TCP_TIMEOUT (TT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgment of data transmission. If an acknowledgment is not received within the specified period, TCP aborts the connection. A small value (less than a few seconds) might cause frequent and unnecessary loss of service during periods of network congestion. A large value might leave users waiting a long period of time after a host or network has failed. If INFINITE is entered, the connection will never abort.

INACTIVITY_TIMEOUT (IT)

Specifies the interval (in seconds) between inactivity checks. If a connection has been idle for the specified time, the gateway sends a TELNET status request to the remote TELNET to determine if the connection is still usable.

Responses

Server TELNET gateway <gateway_name> is changed.

--ERROR-- Server TELNET gateway <gateway_name> is not defined.

Examples

```
senc c='change_server_telnet_gw gateway_name=gw_to_cyber ..
title=ivt_gateway max_connections=5',s=ndi5

Server TELNET gateway GW_TO_CYBER is changed.
```

CHANGE_SOURCE_LOG_GROUP (CHASLG)

Purpose Changes the log messages defined for log group to which a DI's Dependent Log ME belongs. All log message numbers specified by the ADD_MESSAGE_NUMBER parameter will be defined and then all log message numbers specified by the DELETE_MESSAGE_NUMBER parameter will be cancelled. Changes made by this command remain in effect until the next DI reload.

Format **CHANGE_SOURCE_LOG_GROUP**
LOG_GROUP = list 1..1 of name
ADD_MESSAGE_NUMBER = list 1..63 of range 1..32999
DELETE_MESSAGE_NUMBER = list 1..63 of range 1..32999

Parameters *LOG_GROUP (LG)*

Specifies the log group changed by this command. This is the log group to which the defined log messages belong. For this release of CDCNET, only one source log group can be defined per DI. The default log group name is CATENET.

ADD_MESSAGE_NUMBER (AMN)

Specifies one or more log message numbers to be defined for the log group specified. For log messages and their numbers, refer to the CDCNET Diagnostic Messages manual.

DELETE_MESSAGE_NUMBER (DMN)

Specifies one or more log message numbers to be cancelled for the log groups specified. For log messages and their numbers, refer to the CDCNET Diagnostic Messages manual.

Responses Source log group changed.

--WARNING-- No message numbers specified.

--ERROR-- Source log group <name> is not defined.

Examples

```
send_command c='change_source_log_group amn=(40,346,500)',s=tdi3

Source log group changed.

send_command c='change_source_log_group dmn=346',s=tdi3

Source log group changed.
```

CHANGE_SYSTEM (CHAS)

Purpose

Changes memory and buffer allocation boundaries for a DI's memory management functions. For DIs connected to a NOS host, this command also changes whether the DI broadcasts Routing Management Entity (ME) protocol data units, or provides the master clock for the catenet through the Independent Clock ME.

If CHANGE_SYSTEM is included in a system configuration file, a DEFINE_SYSTEM command must precede it. Using NETOU, however, you can enter a CHANGE_SYSTEM command for a DI that does not have a DEFINE_SYSTEM command in its configuration file.

Most changes will be in effect immediately. However, two parameters that are only effective at the next system reload: DATA_BUFFER_SIZE and RESERVED_SYSTEM_SPACE. All changes remain in effect when a DI is reloaded, and stay in effect until you change them again.

Format

CHANGE_SYSTEM
DATA_BUFFER_SIZE = 64..2304
BUFFER_BOUNDARY_PERCENTAGE list 3 of 1..99
MEMORY_BOUNDARY_PERCENTAGE list 3 of 1..99
MEMORY_MANAGER_PERIOD = 1..10
RESERVED_SYSTEM_SPACE = 1000..32768
STANDARD_STACK_SIZE = 0800(16)..2000(16)
DEFAULT_CHANNEL_TRUNK = name
ROUTING_SYSTEM = boolean
CLOCKING_SYSTEM = boolean

Parameters

DATA_BUFFER_SIZE (DBS)

Size, in bytes, of the system data buffers. Parameter value is stored in battery-backed RAM and the effects are not realized until a manual reset or reset command occurs.

The actual buffer size generated is adjusted to be a multiple of a descriptor buffer. The following table defines the actual buffer sizes generated for ranges of entered data buffer size values.

DBS Value	Buffer Size	DBS Value	Buffer Size
64..70	68	1173..1210	1208
71..108	106	1211..1248	1246
109..146	144	1249..1286	1284
147..184	182	1287..1324	1322
185..222	220	1325..1362	1360
223..260	258	1363..1400	1398
261..298	296	1401..1438	1436
299..336	334	1439..1476	1474
337..374	372	1477..1514	1512
375..412	410	1515..1552	1550
413..450	448	1553..1590	1588
451..488	486	1591..1628	1626
489..526	524	1629..1666	1664
527..564	562	1667..1704	1702
565..602	600	1705..1742	1740
603..640	638	1743..1780	1778
641..678	676	1781..1818	1816
679..716	714	1819..1856	1854
717..754	752	1857..1894	1892
755..792	790	1895..1932	1930
793..830	828	1933..1970	1968
831..868	866	1971..2008	2006
869..906	904	2009..2046	2044
907..944	942	2047..2084	2082
945..982	980	2085..2122	2120
983..1020	1018	2123..2160	2158
1021..1058	1056	2161..2198	2196
1059..1096	1094	2199..2236	2234
1097..1134	1132	2237..2274	2272
1135..1172	1170	2275..2304	2310

BUFFER_BOUNDARY_PERCENTAGES (BBP)

Percentages of available buffers corresponding to boundaries between different states of DI buffer availability. The DI dynamically maintains the state of available buffers. The four defined buffer states are: GOOD, FAIR, POOR and CONGESTED.

Specify a list of three integers that specify the three boundaries between the four buffer states. The first value defines the boundary value between GOOD and FAIR; the second value defines the boundary between FAIR and POOR; the third value defines the boundary between POOR and CONGESTED. Values must be listed from highest value to lowest, and differ by at least 5.

MEMORY_BOUNDARY_PERCENTAGES (MBP)

Percentages of available memory that correspond to boundaries between different states of DI memory availability. The DI dynamically maintains the state of available memory. The four defined memory states are: GOOD, FAIR, POOR and CONGESTED.

Specify a list of three integers that specify the three boundaries between the four memory states. The first value defines the boundary value between GOOD and FAIR; the second value defines the boundary between FAIR and POOR; the third value defines the boundary between POOR and CONGESTED. Values must be listed from highest to lowest, and differ by at least 5.

MEMORY_MANAGER_PERIOD (MMP)

Interval, in seconds, that the DI memory manager executes to maintain the DI buffer and memory state.

RESERVED_SYSTEM_SPACE (RSS)

Number of bytes to be reserved in the free memory pool for executive internal allocations. If specified as an odd value, this parameter is rounded up to the next even value.

STANDARD_STACK_SIZE (SSS)

Size, in bytes, of the task's stack size when the initiator of the task does not specify a stack size to the executive. If specified as an odd value, this parameter is rounded to the next even value.

DEFAULT_CHANNEL_TRUNK (DCT)

Specifies the default channel trunk to be used for the configuration of the NOS Network Products interface, gateways, and network management entities that use NOS services. If a default channel trunk is not specified and the DI was loaded across an MCI interface, the trunk over which the DI was loaded becomes the default channel trunk. If a default channel trunk is not specified and the DI was not loaded across an MCI interface, the default channel trunk for the DI is not defined.

ROUTING_SYSTEM (RS)

This parameter is used only in DIs that are supported by NOS hosts. For this release of CDCNET, the feature that uses this parameter is not supported, and the value of this parameter is always FALSE.

CLOCKING_SYSTEM (CS)

Indicates that this DI is to contain the master clock that specifies the date and time for the network. All other DI clocks set their date and time according to this master clock. Default value is FALSE. For DIs connected to a NOS host, there must be only one clocking system DI defined in the catenet with CLOCKING_SYSTEM=TRUE. For DIs supported by NOS/VE hosts, this parameter is not needed, since the DIs obtain the master clock from the NOS/VE host rather than from a clocking system DI. The value of this parameter for an MDI/MTI connected to a NOS/VE host should be FALSE.

Responses

Change of system accepted.

--WARNING-- Change of system accepted. System was not the master clock.

--WARNING-- Change of system accepted. System was already the master clock.

--WARNING-- Change of system accepted. Power on reset <P1> used, please correct.

--ERROR-- Buffer_boundary_percentages values not decreasing or do not differ by 5. The buffer boundary percentages are = (<P1>,<P2>,<P3>).

--ERROR-- Memory_boundary_percentages values not decreasing or do not differ by 5. The memory boundary percentages are = (<P1>,<P2>,<P3>).

--ERROR-- System is not yet defined.

--ERROR-- There is already a master clock in catenet. Network Id: xxxxxx, System Id: xxxxxxxxxxxx.

--FATAL-- The system could not be started as master clock.

Remarks

Proceed with caution if you use values other than the default values for any of the memory management parameters (DATA_BUFFER_SIZE through STANDARD_STACK_SIZE). Changing these values may improve system performance, but can significantly degrade performance as well.

Examples

```
senc c='change_system mbp=(70,80,90)',s=mdi_4

Change of system accepted.
```

CHANGE_TCP_INTERFACE (CHATI)

Purpose

Changes the operational parameters for TCP (DOD's Transmission Control Protocol). Changed values that are negotiated at the beginning of a connection only affect new connections. All other changes occur immediately for all connections. Only parameters that are specified by this command are changed.

Format

CHANGE_TCP_INTERFACE
ACCEPT_STRATEGY = keyword value
ACK_PERCENTAGE = 0..100
MAX_BUFFERS = 1..65535
MAX_SEGMENT_SIZE = 1..4096
MAX_CONNECTIONS = 0..65535
QUIET_TIME = 0..10000
RETRANSMIT_STRATEGY = keyword value
RETRANSMIT_TIME = 0..65535
SECURITY_CHECKING = keyword value
TIME_TO_LIVE = 0..255

Parameters

ACCEPT_STRATEGY (AS)

Specifies the TCP segment accept strategy to be used. The following keyword values are allowed:

Keyword Value	Description
IN_ORDER (IO)	Segments are accepted only in the exact order they are expected. All other segments are discarded. Using this parameter may cause performance degradation and increase the number of retransmitted segments.
IN_WINDOW (IW)	Segments are accepted if they fall within the current TCP window. All other segments are discarded.

Default is IN_WINDOW.

ACK_PERCENTAGE (AP)

Specifies the percentage of the receive window that must be full before an acknowledgment is issued. The default is 50.

MAX_BUFFERS (MB)

Specifies the maximum number of data bytes that TCP holds for a connection for both directions of travel. The default value is 2048 bytes.

MAX_SEGMENT_SIZE (MSS)

Specifies the maximum segment size in bytes to be negotiated for each new connection. The default value is 536 bytes.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous TCP connections. If INFINITE is entered, no restriction is placed on the number of connections. The default value is 200 connections.

QUIET_TIME (QT)

Specifies the number of seconds that TCP must wait, after a connection has closed, before a connection with the same source and destination socket addresses can be opened again. The default value is 120 seconds.

RETRANSMIT_STRATEGY (RS)

Specifies the TCP segment retransmission strategy to be used. The following keyword values are allowed:

Keyword Value	**Description**
BATCH (B)	All unacknowledged segments are retransmitted when the retransmission timer expires.
FIRST_ONLY (FO)	Only the first segment of a sequence of unacknowledged segments is retransmitted when the retransmission timer expires.
ADAPTIVE (A)	Each connection starts in FIRST_ONLY mode. If a subsequent retransmission sequence causes TCP to perform batch retransmission as a series of retransmissions, then TCP switches to BATCH mode. This case detects the instance where the peer TCP is using an IN_ORDER accept strategy.

Default is ADAPTIVE.

RETRANSMIT_TIME (RT)

Specifies the initial number of seconds that TCP should wait for an acknowledgment before retransmitting a data segment. This value changes for an active connection as the actual round-trip time is learned. The default value is 3 seconds.

SECURITY_CHECKING (SC)

Specifies the security checking to be performed on all segments. The following keyword values are allowed:

Keyword Value	**Description**
NONE (N)	The security option supplied in IP datagrams is ignored.
USER_SPECIFIED (US)	The security option specified by the upper layer protocol in the passive or active connect request establishes the security level of the connection.
LEVEL_U (LU)	All connections must be at security level UNCLASSIFIED.

Keyword Value	Description
LEVEL_C (LC)	All connections must be at security level CONFIDENTIAL.
LEVEL_E (LE)	All connections must be at security level EFTO.
LEVEL_M (LM)	All connections must be at security level MMMM.
LEVEL_P (LP)	All connections must be at security level PROG.
LEVEL_R (LR)	All connections must be at security level RESTRICTED.
LEVEL_S (LS)	All connections must be at security level SECRET.
LEVEL_T (LT)	All connections must be at security level TOP SECRET.

If a security level is specified, all connections and all segments received on a connection must match that security level. Any data segments that do not match the security level for a connection are discarded. The default value is NONE.

TIME_TO_LIVE (TTL)

Specifies the Internet Protocol (IP) time-to-live field used by TCP. This is a hop count that is decremented at each gateway traversed by a datagram. When the count in a datagram reaches zero, the datagram is discarded to prevent looping. The default value is 60 hops.

Responses

TCP Options changed.

--WARNING-- Maximum number of connections cannot be changed. The maximum number of connections must be greater than or equal to the number of active connections. If other parameters have been specified, they have been changed.

--ERROR-- TCP is not defined.

Examples

```
senc c='change_tcp_interface accept_strategy=in_window ..
ack_percentage=75 max_buffers=512 max_connections=INFINITE',s=ndi1

TCP options changed.
```

CHANGE_USER_TELNET_GW (CHAUTG)

Purpose

Changes operational parameters of a user TELNET gateway. The original values for these parameters were defined by the DEFINE_USER_TELNET_GW command. Any changes specified by this command affect only new connections to the gateway; existing connections are not affected.

Format

CHANGE_USER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = name
MAX_CONNECTIONS = 0..65535 or *keyword value*
SOURCE_IP_ADDRESS = list of 1..4 of 0..255
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..2147483647
TCP_TIMEOUT = 0..65535 or *keyword value*
INACTIVITY_TIMEOUT = 0..65535

Parameters

GATEWAY_NAME (GN)

The logical name of the user TELNET gateway used in subsequent commands that reference the gateway.

IP_ADDRESS (IA)

The IP address of the host which provides the TELNET interactive service. This user TELNET gateway establishes a connection using this IP address as the destination address. The format is similar to the decimal octet convention used by the TCP/IP community, except that the periods are replaced with commas, and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

TITLE or *TITLES (T)*

Specifies the title(s) by which this gateway service can be accessed. For example, this is the name that CDCNET terminal users supply in the CREATE_CONNECTION command.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the CDCNET catenet that can access this service.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous connections to be supported bythe gateway. If INFINITE is entered, there is no restriction to the number of connections allowed.

SOURCE_IP_ADDRESS (SIA)

Specifies the IP address of the source host to be used by this gateway. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

TCP_PORT_NUMBER (TPN)

Specifies the TCP port number to be used by the gateway. User TELNET issues a TCP active_connect request using the well-known port for the destination port.

TCP_ALLOCATE_SIZE (TAS)

Specifies the amount of data that the gateway queues for each connection. Larger values may improve user response time, especially for PC users (with a standard protocol such as XMODEM), but can increase the number of instances of DI congestion.

CAUTION

Specifying this value is discouraged, and should be done with caution, as poor network service results.

TCP_TIMEOUT (TT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgment of data transmission. If an acknowledgment is not received within the specified period, TCP aborts the connection. A small value (less than a few seconds) might cause frequent and unnecessary loss of service during periods of network congestion. A large value might leave users waiting a long period of time after a host or network has failed. If INFINITE is entered, the connection will never abort.

INACTIVITY_TIMEOUT (IT)

Specifies the interval (in seconds) between inactivity checks. If a connection has been idle for the specified time, the gateway sends a TELNET status request to the remote TELNET to determine if the connection is still usable.

START (S)

Specifies that the newly configured gateway is to be started after it is defined.

Responses

User TELNET gateway <gateway_name> is changed.

--ERROR-- User TELNET gateway <gateway_name> is not defined.

Examples

```
senc c='change_user_telnet_gw gateway_name=gw_to_vax ..
title=(telnet_vax, telnet_unix) max_connections=10',s=ndi1

User TELNET gateway GW_TO_VAX is changed.
```

DISPLAY Commands

DISPLAY_COMMAND_INFORMATION (DISCI)

Purpose

Displays parameter information of the specified command. The specified command must be one of the available network commands. The information includes parameter names, their types, and their default values.

Format

DISPLAY_COMMAND_INFORMATION
COMMAND = command name

Parameters

COMMAND (C)

Specifies the command for which the parameters are to be displayed. You must provide either the full name or the command abbreviation. The specified command must be a network command.

Responses

List of parameter names, parameter abbreviations, and parameter syntax for the specified command. (See example.)

--ERROR-- <string> is not a command.

--FATAL--The command information for the command; <string> cannot be processed.

Examples

```
senc c='display_command_information command=display_hardware_status'

display_name, dn : list  1..30 of name = all
display_option, do : key summary, s, expanded, e = summary
```

DISPLAY_COMMAND_LIST (DISCL)

Purpose Displays a list of network commands for which you are validated. The commands are arranged in alphabetical order. Only the long form of the command is returned. The HELP and DISPLAY_COMMAND_LIST_ENTRY commands are aliases for this command.

Format **DISPLAY_COMMAND_LIST**

Responses <Alphabetical list of network commands. See example.>

Examples

```
senc c='display_command_list'

add_np_gw_outcall                         add_x25_gw_outcall
        .                                         .
        .                                         .
        .
unload_module                             write_terminal_message
```

DISPLAY_COMMAND_LIST_ENTRY (DISCLE)

Purpose The DISPLAY_COMMAND_LIST_ENTRY command is an alias of the DISPLAY_COMMAND_LIST command. Like the DISPLAY_COMMAND_LIST command, this command displays an alphabetical list of network commands.

Format **DISPLAY_COMMAND_LIST_ENTRY**

Responses <Alphabetical list of network commands. See example.>

Examples

```
senc c='display_command_list_entry'

add_np_gw_outcall                      add_x25_gw_outcall
        .                                      .
        .                                      .
        .
unload_module                          write_terminal_message
```

DISPLAY_DATE_AND_TIME (DISDAT)

Purpose Displays the current date and time that is maintained by the DIs to which you send this command.

Format **DISPLAY_DATE_AND_TIME**
DATE_FORMAT = keyword value
TIME_FORMAT = keyword value

Parameters *DATE_FORMAT (DF)*

Specifies how date information is to be displayed. Allowed keyword values include the following, using as an example a date of November 1, 1986, and dd for day, mm for month, and yy for year.

Keyword Value	Description
MDY	Date formatted as mm/dd/yy, as in 11/01/86.
DMY	Date formatted as dd/mm/yy, as in 01/11/86.
ISO	Date formatted as yyyy-mm-dd, as in 1986-11-01

Default value is DMY.

TIME_FORMAT (TF)

Specifies how time information is to be displayed. Allowed keyword values include the following, using as an example a time of 2:41 PM, and hh for hour, mm for minute, ss for second, and XX for AM or PM identifier.

Keyword Value	Description
AMPM	Time formatted as hh:mm XX, as in 2:41 PM.
HMS	Time formatted as hh:mm:ss, as in 14:41:38.

Default value is HMS.

Responses System date and time
(Followed by date and time in selected format. See example.)

Examples

```
senc c='display_date_and_time',s=main_mdi

System date and time
    14/10/86 15:09:24

senc c='display_date_and_time df=mdy,tf=ampm',s=mti_83

System date and time
    02/20/86 10:36 AM.
```

DISPLAY_DEVICE_OUTCALL_STATUS (DISDOS)

Purpose Displays the current status of the Device Outcall Service.

Format **DISPLAY_DEVICE_OUTCALL_STATUS**

Responses Device Outcall Service Defined and Started.

Device Outcall Service Not Defined

Examples

```
senc c='display_device_outcall_status',s=tdi3

Device Outcall Service Defined and Started.
```

DISPLAY_DIRECTORY_STATUS (DISDS)

Purpose Displays the operating status of the Directory Management Entity (ME) in a DI. The command supports summary, expanded, and detailed displays.

Format **DISPLAY_DIRECTORY_STATUS**
DISPLAY_OPTION = list 1..3 of keyword value
TITLE = list 1..15 of name or keyword value

Parameters *DISPLAY_OPTION (DO)*

Selects the level of status for the directory status display. The following keyword values are allowed.

Keyword Value	Description
SUMMARY (S)	Selects the summary display.
EXPANDED (E)	Adds the expanded information to the display. Refer to the status display description following the examples for more information.
DETAIL (D)	Selects a detailed display. Refer to the status display description following the examples for more information.
ALL (A)	Selects all displays.

Default is SUMMARY.

TITLE (T)

A list of one or more titles for which expanded or detailed information is desired. Enter the title as a string value within apostrophes (') (see examples). This parameter is meaningful only if you choose an expanded or detailed display with the DISPLAY_OPTION parameter. If you select a summary display, this parameter is ignored. The default value for this parameter is ALL (display information for all titles).

Responses Directory Status
(Followed by the status display. See example. If a specified title is not registered, the following response is inserted in the status display):

Title <name> is not registered.

Remarks For more information on Directory Management Entity, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
senc c='display_directory_status',s=mdi1

Directory Status

current registered titles = 4
current received titles = 25
current translation requests = 8

senc c='display_directory_status do=e t=''$I_LOG_ME_LOG_GROUP_1''',..
s=mdi1

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  address = 00001ACB0800251000930 1BC      priority = 3
  registered by: TDI_AVCD                      85/11/14  02:15:32

senc c='display_directory_status do=e t=''TIMESHARING''',mdi1

Directory Status

title = TIMESHARING
  address = 00001ACB0800251000930 1BC        priority = 3
  registered by: CYBER 180 SN00312, CLASS C     85/11/14  02:15:32

senc c='display_directory_status do=e t=''$I_LOG_ME_LOG_GROUP_2''',..
system=mdi1

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  address = 00001ACB0800251000930 1BC      priority = 3
  registered by: $DI_080025011312              85/11/14  02:15:32

senc c='display_directory_status do=d t=''$I_LOG_ME_LOG_GROUP_1''',..
system=mdi1

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  community =
  user_info =
  address = 00001ACB0800251000930 1BC
  priority = 3
  service = GENERIC_TRANSPORT
  translation_domain = CATENET
  class = INTERNAL
  dirid = 00001ACB08002510009385111402153 21920
```

Directory Status Display Description

The DISPLAY_DIRECTORY_STATUS command supports summary, expanded, and detail displays.

The summary display returns the following information.

Current number of titles registered by the system

Current number of titles received from other systems

Number of active title translation requests for the system

The expanded display returns the following information for each title displayed.

Title

Address

Title priority

System that registered the title

Date and time of registration

The detail display returns the following for each title displayed. This information is derived from the directory registration control block and the Directory ID for each title.

title	The title for which information is displayed.
community	Title registration domain.
user_info	Information saved by the title registrator.
address	The title's network, system, service access point (SAP), entry point, or procedure address.
priority	Hierarchical priority level assigned to duplicates of a title. Priority is established by the registrator of the title.
service	The layer connection service used by the registrator of the title. Services defined include the following: UNKNOWN XNS_INTERNET XNS_TRANSPORT GENERIC_TRANSPORT SESSION VIRTUAL_TERMINAL BATCH_TRANSFER CDC_DEFINED_XXX CUSTOMER_DEFINED_XXX
translation_domain	Where the title may be translated. Possible values include CATENET or LOCAL_SYSTEM.

`class`	EXTERNAL, visible to CDCNET users, or INTERNAL, hidden from CDCNET users.
`dirid`	System name where the title was registered, plus the date and time it was registered.

DISPLAY_DI_SYSTEM_STATUS (DISDSS)

Purpose

Returns general information about the operation of a DI and resource usage in the DI, such as date and time of the last reload, version of load file used, states of buffers and memory, and CPU usage. An expanded status display also includes the responses to the DISPLAY_HARDWARE_STATUS, DISPLAY_LINE_STATUS, and DISPLAY_NETWORK_STATUS commands.

Format

DISPLAY_DI_SYSTEM_STATUS
DISPLAY_OPTION = keyword value

Parameters

DISPLAY_OPTION (DO)

Selects a summary or expanded status response. There are two possible values for this parameter.

Keyword Value	Description
SUMMARY (S)	Selects general DI operating system status and does not include the additional hardware, line, and network status displays.
EXPANDED (E)	Selects general system status and status for the hardware component(s) in the DI. The hardware display is a combination of the hardware status, line status, and network status displays, and is appended to the end of the summary display.

Default is SUMMARY.

Responses

DI System Status.
(Followed by status display. See example.)

Examples

In this example, the DISPLAY_DI_SYSTEM_STATUS command is entered, omitting the DISPLAY_OPTION parameter. The command returns a summary status response.

```
senc c='display_di_system_status',s=MTI_83

DI System Status
system name = MTI_83
system address = 080025100083(16)
boot version number = 1511(16)
software release level = 1511(16)
number of tasks = 64
free SMM memory = 445490
percent CPU utilization = 4
buffer state = good
memory state = good
date and time of last reload = 86/04/27  11:23:45

Buffer Status
type        total buffers       available buffers        buffer size
data        4216                3820                     144
descriptor  1436                1394                     32

SMM Memory Status
total memory    available memory    extents     deloadable memory
1572864         279752              55          119816

PMM Memory Status
total memory    available memory    extents     deloadable memory
131072          31500               9           0

MPB RAM Status
total memory    available memory    extents     deloadable memory
16384           1902                1           0
```

DI System Status Display Description

The DI System Status Display includes general DI operating system information, buffer and memory usage status and, optional hardware, line, and network status displays. For descriptions of the hardware, line, and network status displays, refer to the commands that generate those status displays.

The general DI information section includes:

`system name`	The DI's name, assigned during configuration.
`system address`	The DI's unique address.
`boot version number`	Version number of the boot file currently loaded in and running on the DI. Taken from exception list or INITMDI.
`software release level`	Version number of the compiled software currently loaded in and running on the DI. This value is defined in a common deck and indicates the released version level.
`number of tasks`	Amount of work that has been scheduled for the DI's Central Processing Unit (CPU) to perform. Tasks that are scheduled may not actually be executing.
`free SMM memory`	Amount of memory on the SMM board that is not currently assigned to a software process.
`free PMM memory`	Amount of memory on the PMM board that is not currently assigned to a software process.
`percent CPU utilization`	Percentage of time the CPU on the MPB board is performing work as opposed to being idle.
`buffer state`	Describes level of buffer availability. The four states of buffer availability are GOOD, FAIR, POOR and CONGESTED. Refer to the BUFFER_BOUNDARY_PERCENTAGES parameter in the DEFINE_SYSTEM command description. Each boundary is expressed as a percentage of total resources allocated after the DI is configured.
`memory state`	Describes level of memory availability. The four levels of memory availability are GOOD, FAIR, POOR, and CONGESTED. Refer to the MEMORY_BOUNDARY_PERCENTAGES parameter in the DEFINE_SYSTEM command description. Each boundary is expressed as a percentage of total resources allocated after the DI is configured.
`date and time of last reload`	The time when the DI software was completely reloaded.

Buffer Status	Displays the following information:	
	Total Buffers	The total number of buffers allocated for use by the DI.
	Available Buffers	The number of allocated numbers that are now currently in use.
	Buffer Size	The size, in bytes of a particular buffer.
Memory Status (PMM, SMM, MPB)	Displays the following information:	
	Total Memory	The total number of bytes of memory for this DI.
	Available Memory	The total number of bytes of memory available for loading modules and allocating structures by these modules.
	Extents	The number of memory fragments into which available memory is divided.
	Reloadable Memory	The number of bytes that can be used when a deloadable threshold is reached. Deloadable memory is made up of nonactive tasks.

For expanded status displays only, the remainder of the display is a summary status of the various DI components in the DI, network solutions, and communication lines. For specific information about these status entries, see the other display status commands (DISPLAY_HARDWARE_STATUS, DISPLAY_LINE_STATUS, DISPLAY_NETWORK_STATUS).

DISPLAY_FILE_SUPPORT (DISFS) (NOS Only)

Purpose Displays the file types supported by an Independent File Access ME residing on a NOS MDI or MTI. The display also indicates the status of the Independent File Access ME connection to the host through the trunk. If the status is DOWN, the connection is down. If the status is ACTIVE, the connection is up and in use.

Format **DISPLAY_FILE_SUPPORT**
TRUNK_NAME = name

Parameters *TRUNK_NAME (TN)*

Displays the trunk name of the logical link which used to support the file access connection. If this parameter is not specified, the default trunk is used. The default channel trunk name is specified by a DEFINE_SYSTEM command.

Responses File Types supported for trunk <name>. Connection is <state>. (Followed by the DISFS display (see example). Within the status display, the following responses will replace the display response if file support is not defined for a specified trunk or if no file support is defined for the system).

File Support is not defined for trunk <name>.

File Support is not defined for the system.

Remarks For more information on File Access Management Entity, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
senc c='display_file_support',s=mdi2

File Types supported for trunk 03.  Connection is active.

USER_PROCEDURE
TERMINAL_PROCEDURE
LOAD_PROCEDURE
LIBRARY
EXCEPTION
DUMP
CONFIGURATION
BOOT
```

DISPLAY_HARDWARE_STATUS (DISHS)

Purpose

Displays status of the processor, peripheral, logic, memory, and line control boards in a DI (large boards: MPB, MCI, ESCI, CIM, PMM, SMM; and small boards: LIM and URI boards). The command displays status for all boards or a set of boards specified by their physical names. If no parameters are supplied with the command, the status of all large boards and all LIM and URI boards are displayed. If a board or port's physical name is entered, the status of the board or port of that name is displayed.

This command supports two levels of display: summary and expanded. A summary display includes the status of large boards, LIMs and URIs (if no device names are entered), or the status of boards specified by device name with the command. An expanded display includes the summary display information plus the status of all subassemblies to a board, for example, LIM or URI boards controlled by a named CIM board. Hardware status displays are described following the command examples.

Format

DISPLAY_HARDWARE_STATUS
DEVICE_NAME = list of name
DISPLAY_OPTION = keyword value

Parameters

DEVICE_NAME (DN)

The physical name of the device for which status is to be returned (see Physical Names, chapter 1). This parameter is optional and has no default value.

DISPLAY_OPTION (DO)

Specifies level of status display. There are two possible values for this parameter.

Keyword Value	Description
SUMMARY (S)	Provides status of large boards, LIMs, and URIs (if no device names are entered), or status of boards you specifically select using the DEVICE_NAME parameter.
EXPANDED (E)	Includes the summary display information plus the status of all subassemblies to a board (such as LIM ports), or boards controlled by a card specified by the DEVICE_NAME parameter (such as LIM or URI boards controlled by a CIM board).

Default is SUMMARY.

Responses

Hardware Status
(Followed by the requested hardware status. See Example.)
Within the status display, the following responses will be inserted if a device name is unknown or if the device is not installed.

Device name <name> unknown or not installed.

Examples This example shows summary status for all boards in a TDI. Board slot 3 is empty.

```
senc c='display_hardware_status',s=tdi_1

Hardware Status
device name    state    status          version   lim/bank/port  type
$MPB0          on       configured      0000(16)
$PMM1          on       configured      0008(16)
$SMM2          on       configured      0000(16)   2
    3          off      not config.     0000(16)
$MCI3          on       protocol mism.
$CIM4          on       configured      0000(16)   0,1,2,3
$CIM5          down     not config.     0000(16)
$ESCI6         on       active          0010(16)
$MCI7          on       active          0000(16)
$LIM0          on       enabled                    4             RS232
$LIM1          down     configured                 4             RS232
$LIM2          on       enabled                    2             RS449
$LIM3          on       configured                 2             RS449
$URI4          on       enabled
```

This example shows the summary status display for a LIM.

```
senc c='display_hardware_status dn=$lim4_port0',s=tdi_2

Hardware Status
                Port Status
device name    state    status    version   lim/bank/port   type
$LIM4_PORT0    on       enabled                             async
```

This example shows the summary status display for a MPB.

```
senc c='dishs dn=$mpb0',s=mti_83

Hardware Status
device name    state    status    version   lim/bank/port   type
$MPB0          on       configured 0000(16)
```

This example shows the expanded status display for all LIMs on the default mdi.

```
senc c='dishs do=expanded'

Hardware Status
device name    state    status      version    lim/bank/port    type
$LIM0          on       configured             4                RS232
$LIM1          on       configured             4                RS232
$LIM2          on       configured             4                RS232
$LIM3          on       configured             4                RS232
$LIM4          on       configured             4                RS232
$LIM5          on       configured             4                RS232
$LIM6          on       configured             4                RS232
$LIM7          on       configured             4                RS232
$LIM0_PORT0    on       enabled                                 ASYNC
$LIM0_PORT1    on       enabled                                 ASYNC
$LIM0_PORT2    on       enabled                                 ASYNC
$LIM0_PORT3    on       enabled                                 ASYNC
$LIM1_PORT0    on       enabled                                 ASYNC
$LIM1_PORT1    on       enabled                                 ASYNC
$LIM1_PORT2    on       enabled                                 ASYNC
$LIM1_PORT3    on       enabled                                 ASYNC
$LIM2_PORT0    on       enabled                                 ASYNC
$LIM2_PORT1    on       enabled                                 ASYNC
$LIM2_PORT2    on       enabled                                 ASYNC
$LIM2_PORT3    on       enabled                                 ASYNC
$SMM2_BANK0    on       configured             2
$SMM2_BANK1    on       configured             2
```

Hardware Status Display Description

The hardware status display describes each board as follows.

device name
: The physical name of the board or LIM port, specified as $board type_slot number, as in $MPB0, $PMM1, and $CIM4. An empty board slot for a major board is assigned the slot number. For example, in the first command example, the third major board slot is empty, and has the name 3.

state
: Operational state of the board, which may be:

on	Operational; available for use by the communications system.
down	Not operational; available for diagnostic tests only.
off	Not operational or not installed; not available for use without intervention such as installing boards and changing the board's operational state by the CHANGE_ELEMENT_STATE command.

status
: Indicates how the board is being used by the DI's communications system. A board may have one of the following status conditions.

not avail.	The port exceeds the 48 port limit for LIM ports connected to one CIM board and is thus unavailable for use.
configured	Board has been configured (prepared) for use by the communications system.
not config.	Board is not configured for use by the communications system.
enabled	Board is configured, and is in use by the communications system.
active	Active communications are being carried over the device. Appropriate communications protocols are being exchanged.
protocol mismatch	MCI cannot support protocol version requested by PIP. Reflects status of MCI board.

version
: The current hardware version of the board (not applicable to ports).

`lim/bank/port` This section gives information about the different types of boards and any subordinate DI hardware that a board controls. This status information is provided under the following headers.

`lim` The LIM and URI boards a CIM board controls.

`bank` The number of memory banks on an SMM board.

`port` The number of ports defined for a LIM board.

`type` For LIMs, this field describes the the physical connection type on LIM board, such as RS-232 and RS-449. For ports, this field describes the terminal interface program (TIP) controlling the port, such as the asynchronous TIP. Compare the information under the Type column in the first and second examples to see how information in the Type column differs between LIMs and ports.

DISPLAY_HDLC_NET_OPTIONS (DISHNO)

Purpose

Displays the configuration of an HDLC network. You can display the configuration of a single HDLC network, or you can display the configuration of all the HDLC networks defined for the specified system. Address the network by its logical name.

HDLC network option displays are described following the command examples.

Format

DISPLAY_HDLC_NET_OPTIONS
NETWORK_NAME = list of 1 .. 15 name
DISPLAY_OPTION = list of keyword value

Parameters

NETWORK_NAME (NN)

Logical name of an HDLC network assigned by a DEFINE_HDLC_NET command.

DISPLAY_OPTION (DO)

Selects one or more network attributes for display. The default is ALL. The following display options, described in the DEFINE_HDLC_NET command at the end of this chapter, are allowed.

TRUNK_NAME (TN)
NETWORK_ID (NI)
NETWORK_NAME (NN)
COST(C)
RELAY_ALLOWED (RA)
ROUTING_INFO_NETWORK (RIN)
OUTPUT_QUEUE_LIMIT (OQL))
ALL

Responses

HDLC Network options
(Followed by the display of the HDLC network options. See example.) Within the options display, the following responses are inserted if NETWORK_NAME is not defined, is not an HDLC network, or no HDLC networks are defined.

Network <name> is not defined.

Network <name> is not an HDLC network.

No HDLC networks are defined for this system.

Examples

This command returns a list of the options selected for the specified HDLC network.

```
senc c='display_hdlc_net_options nn=hdlc network 2'

HDLC Network options
trunk_name = HDLC_TRUNK_LINE_1
network_id = 123456(16)
network_name = HDLC_NETWORK_2
cost = 2000
relay_allowed = yes
routing_info_network = yes
output_queue_limit = 30000
```

DISPLAY_HDLC_TRUNK_OPTIONS (DISHTO)

Purpose

Displays the configuration of HDLC trunks. You address an HDLC trunk by its trunk name. If you enter no trunk name, the display presents the configuration of all HDLC trunks defined for the system.

Format

DISPLAY_HDLC_TRUNK_OPTIONS
TRUNK_NAME = list 1 .. 15 of name
DISPLAY_OPTION = list of keyword value

Parameters

TRUNK_NAME (TN)

Logical name of an HDLC trunk. Name assigned by a DEFINE_HDLC_TRUNK command.

DISPLAY_OPTION (DO)

Selects one or more trunk attributes for display. The following display options, described in the DEFINE_HDLC_TRUNK command at the end of this chapter, are allowed.

LIM (L)
PORT (P)
LOCAL_ADDRESS (LA)
REMOTE_ADDRESS (RA)
TRUNK_NAME (TN)
OPTIONS (O)
MAX_UNACK_FRAMES (MUF)
SREJE_QUEUE_SIZE (SQS)
MAX_FRAME_SIZE (MFS)
PF_RECOVERY_TIMER (PRT)
ERROR_RECOVERY_TIMER (ERT)
RETRANSMISSION_LIMIT (RL)
TRUNK_SPEED (TS)
CLOCKING (C)
INTERACTIVE_BANDWIDTH (IB)
ALL

Default is ALL.

Responses

HDLC Trunk options (see example).
Within the options display, the following responses will be inserted if a TRUNK_NAME is not defined, is not an HDLC trunk, or no HDLC trunks are defined.

Trunk <name> is not defined.

Trunk <name> is not an HDLC trunk.

No HDLC trunks are defined for this system.

Examples

```
senc c='display_hdlc_trunk_options tn=HDLC_TRUNK_1',s=mdi_8a

HDLC Trunk options
lim = 3
port = 1
local_address = 123
remote_address = 85
trunk_name = HDLC_TRUNK_1
options = (SIM_ON,RESET_ON,IFRAME_ON)
max_frame_size = 1500
pf_recovery_timer = 500
error_recovery_timer = 2000
retransmission_limit = 20
trunk_speed = 19200
clocking = transmit
interactive_bandwidth = 7
```

DISPLAY_LINE_OPTIONS (DISLO)

Purpose Displays the configuration of communications line(s) and unit record interface line(s) supported by the terminal interface programs (TIPs).

Format **DISPLAY_LINE_OPTIONS**
LINE_NAME = list of 1..15 names
DISPLAY_OPTION = list of keyword value

Parameters *LINE_NAME (LN)*

Specifies the logical name of one or more lines for display. The line(s) were previously defined by the DEFINE_LINE commands. If you do not specify a LINE_NAME the display includes all lines defined for the system.

DISPLAY_OPTION (DO)

Specifies one or more of the line attributes for display. These attributes are defined by parameters on the DEFINE_LINE command that configured the line. For more information on these parameters, refer to the DEFINE_LINE command description. Allowed keyword values include the following.

Keyword Value	Description
LIM (L)	Displays the line interface module number on the MTI/MDI to which the line is connected.
PORT (P)	Displays the port number to which the LIM is attached.
TIP_NAME (TN)	Displays the type of TIP that serves the line.
LINE_NAME (LN)	Displays the logical name of the line.
LINE_TYPE (LT)	Displays the type of line, whether the line is SWITCHED or DEDICATED.
LINE_SUB_TYPE (LST)	Displays the subtype of line. Further qualifies the line type.
CARRIER_TYPE (CT)	Displays the type of carrier control on the line.
LINE_SPEED (LS)	Displays the line speed of the communication line in bits per second.
AUTO_RECOGNITION (AR)	Displays what type of auto recognition is performed for asynchronous lines.
TRANSMISSION_BLOCK_SIZE (TBS)	Displays the transmission block used for transmission blocks sent to the terminal device(s) on this line.

Keyword Value	Description
CONNECTION_CONNECT_ TIMEOUT (CCT)	Displays how much time the line user has to create the first $input/$output connection. When this parameter is specified on the DEFINE_LINE command, it is rounded up to the nearest multiple of 4 seconds. As a result, there may be a discrepancy between the value specified on the DEFINE_LINE command and the value displayed by this command.
CONNECTION_DISCONNECT_ TIMEOUT (CDT)	Displays how much time the line user has to establish a new $input/$output connection after the last such connection has been disconnected. When this parameter is specified on the DEFINE_LINE command, it is rounded up to the nearest multiple of 4 seconds. As a result, there may be a discrepancy between the value specified on the DEFINE_LINE command and the value displayed by this command.
TERMINAL_DEFINITION_ PROCEDURE (TDP)	Displays the name of a terminal definition procedure (TDP) file.
TERMINAL_USER_PROCEDURE (TUP)	Displays the name of the terminal user procedure (TUP) that defines characteristics of terminals.
USER_CONNECTION_LIMIT (UCL)	Displays the maximum number of connections a user of the line can have outstanding at one time.
EIA_FLOW_CONTROL (EFA)	Displays whether the Clear to Send and Request to Send flow control is used to stop and resume the flow of input and output data.
CLOCKING (C)	Displays whether the LIM internally generates the clock signal for data on this line or uses an externally generated clock signal for data on the line.
DATA_PARITY (DP)	Displays the parity for data received and transmitted on this line.

Default is ALL.

Responses Line options
(Followed by the options display [see example]. Within the options display, the appropriate following response will appear if the LINE_NAME is not defined or if no lines are defined for the system).

Line_name <name> is not defined.

No lines are defined for this system.

Examples

```
senc c='display_line_options line_name = engin_terminal_31'

Line options
lim = 1
port = 2
tip_name = ASYNCTIP
line_name = ENGIN_TERMINAL_31
line_type = dedicated
line_sub_type = LOCAL
carrier_type = constant
line_speed =  9600
auto_recogniton = none
transmission_block_size = 4095
connection_connect_timeout = 100
connection_disconnect_timeout = 50
terminal_definition_procedure =  TDF_ENGIN
terminal_user_procedure = TDU_ENGIN_31
user_connection_limit - 4
eia_flow_control = 4
clocking = internal
data_parity = even
```

DISPLAY_LINE_STATUS (DISLS)

Purpose

Displays operational status of communication lines and URI lines connected to a DI. You may choose status of all lines (by specifying no parameters), lines controlled by specific terminal interface programs (TIPs) (by specifying the TIP's name), or individual lines (by specifying the names of the lines). This command also returns the status of the terminal/batch devices attached to the lines and the status of the connections for the devices in expanded or detailed displays.

If multiple parameters are specified, status is displayed for lines matching the combination of parameter values specified. For example, if you request status by both TIP name and line name, then the status for all enabled lines controlled by the TIP and status for all the lines of the names you specify is displayed. A named line that is also controlled by a named TIP will appear twice in the status display.

Line status displays are described following the command examples.

Format

DISPLAY_LINE_STATUS
LINE_NAME = list 1..7 of name
TIP_NAME = list 1..32 of name
LINE_STATE = list 1..5 of keyword value
DISPLAY_OPTION = list of keyword value

Parameters

LINE_NAME (LN)

Logical name of one or more communication lines for which you are requesting status.

TIP_NAME (TN)

Logical name of the TIP controlling the lines for which you are requesting status.

LINE_STATE (LS)

Selects the lines to display by line state or line states if neither the TIP_NAME nor LINE_NAME parameter is specified, or if only TIP_NAME is specified. The following values are allowed for the LINE_STATE parameter.

Keyword Value	Description
ACTIVE (A)	Selects display of active lines only.
AUTOREC_ACTIVE (AA)	Selects display of lines for which auto recognition of line speed, parity, and/or character set is taking place.
DISABLED (D)	Selects display of disabled lines.
ENABLED (E)	Selects display of active and enabled lines.
LOADING_TIP (LT)	Selects display of lines for which the controlling TIP is being loaded.
ALL	Selects display of all lines in all line states.

Default is ALL.

If you do not specify TIP_NAME or LINE_NAME, all lines that are in the specified line state will be displayed. If you specify TIP_NAME, all lines supported by the specified TIP that are in the selected line state will be displayed. Selecting display by the LINE_NAME parameter overrides selecting display by LINE_STATE. The status of a specific line name is given regardless of the line state specified.

DISPLAY_OPTION (DO)

Selects a level of status response. The following display options are allowed.

Keyword Value	Description
SUMMARY (S)	Selects general line status.
EXPANDED (E)	Selects status of terminal devices connected by the lines.
DETAIL (D)	Selects status of the active connections for the terminal devices connected by the lines.

Default is SUMMARY.

Responses

Line Status.
(Followed by the line status display. See examples.)
Within the status display, the following responses will be inserted if a line name or TIP name is not defined in the DI's logical configuration, or if no lines match the requested line state.

Line_name <name> not defined.

No <line_state> lines found for the <tip_name> tip.

No <line_state> lines found.

No lines defined for the <tip_name> TIP.

No lines defined. No line status to report.

No devices defined.

No connections active.

Examples

This command returns the status of all communication lines connected to a DI named North_TDI_2. The display option is SUMMARY.

```
senc c='display_line_status',s=north_tdi_2

Line Status
line name      line          line     tip      line     physical
               state         type     name     speed    device name
ENGIN_BLD_1    disabled      swt.     async    1200     $LIM0_PORT0
COMPSCI_02     active        ded.     async    9600     $LIM1_PORT0
COMPSCI_03     enabled       ded.     async    9600     $LIM1_PORT1
COMPSCI_04     active        ded.     async    9600     $LIM1_PORT2
COMPSCI_05     loading_tip   ded.     async    9600     $LIM1_PORT3
COMPSCI_06     autorec_act   ded.     async    AUTO     $LIM2_PORT3
```

This command returns the status of a specific line, using the LINE_NAME (LN) parameter.

```
senc c='display_line_status ln=compsci_02',s=north_tdi_2

Line Status

line name      line        line     tip      line     physical
               state       type     name     speed    device name
COMPSCI_02     enabled     ded.     async    9600     $LIM1_PORT0
```

This example requests an expanded status display for two lines.

```
senc c='display_line_status ln=(line01,line10),do=e',s=tdi_4

Line Status

LINE01                          tip name:  ASYNC
  device name: $CONSOLE_100081_00010001 address: 00/01 state: active

LINE10                          tip name:  ASYNC
  device name: $CONSOLE_100081_00010000 address: 00/00 state: active
```

This example requests a detail status display for one line.

```
senc c='display_line_status ln=line01,do=d',s=tdi_4

Line Status

$CONSOLE_100081_00010001            line name: line01
  service name: ARH907                         INTERACTIVE
  input state: off       output state: hold    output queued: 4/2875

> service name: ARH817                         INTERACTIVE
  input state: send      output state: send    output queued: 1/572

$CONSOLE_100081_00010002            line name: line01
> service name: $CDCNET_COMMAND                INTERACTIVE
  input state: send      output state: send    output queued: 0/0
```

Line Status Display Description

The summary line status display information includes:

`line name` — The logical name of the communication line.

`line state` — Operational state of the line, which may be:

`active`	Communications are being carried over the line; appropriate communications protocols are being exchanged.
`deleting`	The line is in the process of being logically deleted.
`enabled`	The line is configured for use by the DEFINE_LINE command, but the line may not be active.
`disabling`	The line is in the process of being disabled.
`disabled`	The line is configured but not enabled for communications by the TIP controlling the line. The line is not started or communications have failed on the line.
`switching`	The line is in the process of being switched.
`downing`	The line is in the process of being disabled. For example, if a STOP_LINE were being sent to a line, the status for the line would be DOWNING. Once the command executed, the status changes to DISABLED.
`reenabling`	In process of being enabled. Periodic retry of communications on a disabled line have succeeded.
`autorec_active`	Auto recognition of speed, parity, and/or character code set is taking place.
`loading_tip`	The controlling TIP for the line is being loaded.

`line type` — Type of line, which may be either switched (swt.) or dedicated (ded.).

`tip name` — The name of the TIP that is controlling the communication line.

`line speed` — Communication line speed in bits per second.

`physical device name` — The physical name of the LIM/Port used for the communication line.

The expanded line status display describes the devices for each line as follows.

line name	The logical name of the communication line.
tip name	The name of the controlling TIP.
device name	The logical name of the device.
address	The physical address (cluster address/device address) of the device.
state	The state of the device, as follows:

active	Communications with the device are active.
inactive	Communications with the device are inactive.
down	The device is down.
stopped	Data transfer for the device has been stopped by the terminal user.
not ready	The device is not ready.

The detail line status display describes the active connections for the interactive and batch devices for each line as follows.

device name	The logical name of the device.
working connection	Indicated by a > character preceding its status.
service name	The logical name of the service to which the device is currently connected. $CDCNET_COMMAND is displayed if no connections are present.
connection type	Type of terminal connection for the line, which may be INTERACTIVE or BATCH.
input state	The input state for the connection, which may be:

active	Input is active.
off	Input is off; the connection is not the working connection.
flow cntl	Transmission of further input stopped due to network flow control.
sync	Input interrupted (for example, a user has entered an interrupt sequence).

`output state` The output state for the connection, which may be:

`send`	Output sent to the device as received.
`hold`	Output held by the network until reenabled by the user.
`discard`	Output discarded until reenabled by the user.
`interrupt`	Output aborted (interrupted) by the user.
`flow cntl`	Transmission of further output stopped due to network flow control.
`sync`	Output interrupted.
`output queued`	The number of messages / number of bytes queued for output.

DISPLAY_LOGICAL_NAMES (DISLN)

Purpose

Returns the logical names for trunks, network solutions, communication lines, gateways, NP interface definitions, X.25 interface definitions, Unit Record Interface lines, batch I/O stations, Network Transfer Facility (NTF) remote systems, NTF batch streams, devices, and TIPs for a specified DI.

Format

DISPLAY_LOGICAL_NAMES
DISPLAY_OPTION = list of keyword value

Parameters

DISPLAY_OPTION (DO)

Specifies one or more types of logical names for display. The following keyword values are allowed.

TRUNK_NAME (TN)
NETWORK_NAME (NN)
LINE_NAME (LN)
GATEWAY_NAME (GN)
INTERFACE_NAME (IN)
TIP_NAME (TIP)
I_O_STATION NAME (IOSN)
DEVICE_NAME (DN)
REMOTE_SYSTEM_NAME (RSN)
STREAM_NAME (SN)
ALL

Default is ALL.

Responses

System logical names.
(Followed by logical names display. See example.)

Examples

```
senc c='display_logical_names',s=engin_tdi

System logical names

Trunk_names
   ETHER_ESCI1

Network_names
   TDI_TRUNK

Line_names
   ENGINEERING_PORT_1      ENGINEERING_PORT_2
   ENGINEERING_PORT_3      ENGINEERING_PORT_4
   ENGINEERING_PORT_5      ENGINEERING_PORT_6
   ENGINEERING_PORT_7      ENGINEERING_PORT_8
   ENGINEERING_PORT_9      ENGINEERING_PORT_10
   ENGINEERING_PORT_11     ENGINEERING_PORT_12
   ENGINEERING_PORT_13     ENGINEERING_PORT_14
   ENGINEERING_PORT_15     ENGINEERING_PORT_16

Gateway_names
   -No gateway_names defined

Interface_names
   -No interface_names defined
```

```
Tip_names
   ASYNCTIP

I_o_station_names
   -No i_o_station_names_defined

Device_names
   ASYNC_TERMINAL_1        ASYNC_TERMINAL_2
   ASYNC_TERMINAL_3        ASYNC_TERMINAL_4
   ASYNC_TERMINAL_5        ASYNC_TERMINAL_6
   ASYNC_TERMINAL_7        ASYNC_TERMINAL_8
   ASYNC_TERMINAL_9        ASYNC_TERMINAL_10
   ASYNC_TERMINAL_11       ASYNC_TERMINAL_12
   ASYNC_TERMINAL_13       ASYNC_TERMINAL_14
   ASYNC_TERMINAL_15       ASYNC_TERMINAL_16

NTF remote_system_names
   -No remote_system_names defined

NTF batch_stream_names
   -No batch_stream_names defined
```

DISPLAY_MEMORY (DISM)

Purpose

Displays the contents of memory, beginning at the machine address you specify. The amount of memory displayed is the product of two parameters, BYTE_COUNT and REPEAT_COUNT that you specify. The command also returns the module name and offset in the module of the displayed address, if the starting address is within a section of a module. The memory display is in hexadecimal and ASCII representation.

Format

DISPLAY_MEMORY
  ADDRESS = 0..0FFFFFF(16) or name
  BYTE_OFFSET = 0..0FFFF(16)
  BYTE_COUNT = 1..1000(16)
  REPEAT_COUNT = 1..1000(16)

Parameters

ADDRESS (A)

Location of the first memory byte you want to display. Enter the name of the module, or the entry point if you want to display memory within a module; otherwise, enter the numeric address of the starting memory location you want to display. This value is considered the base address.

BYTE_OFFSET (BO)

Provides the offset to the base address given by the address parameter. Add the value of the BYTE_OFFSET parameter to the address parameter value, forming the new address. Default value is 0.

BYTE_COUNT (BC)

Specifies the number of bytes in each line to be displayed. Use with the REPEAT_COUNT parameter to specify the number of bytes to display. (Refer to the description of the REPEAT_COUNT parameter below.) The default number of bytes to display is 1. The default number of bytes per line is 16.

REPEAT_COUNT (RC)

Specifies the number of lines to be displayed. Use with BYTE_COUNT to specify the number of bytes to display. The default value when calculating the number of bytes to display is 1.

The number of lines you display will only match the specified REPEAT_COUNT if BYTE_COUNT is greater than 1 or less than or equal to 16. If BYTE_COUNT is 1 or greater than 16, the number of lines displayed will be the number required to display BYTE_COUNT times the REPEAT_COUNT number of bytes at 16 bytes per line.

Responses

Memory displayed (see example).

--WARNING-- Some memory to be displayed is not in a valid address range.
Displayed memory length truncated.

--WARNING-- Bus error encountered, display terminated.

--ERROR-- First address to be displayed is not in a valid address range.

--ERROR--Address xxx not found.

Examples

```
senc c='display_memory address=system_data repeat_count=128'

11808C  SYSTEM_AUDITS+2AA
11808C  0006 4D54 495F 3833 2020 2020 2020 2020     MTI_83
11809C  2020 2020 2020 2020 2020 2020 2020 2020
1180AC  2000 0000 0015 0041 0036 000E 0007 000C         A 6
1180BC  0007 0024 0009 0006 0006 03C4 3E7C 8611     $          >|
1180CC  2018 1348 9600 0000 0001 0800 2510 0083     H          %
1180DC  0000 0000 0100 0023 200A 2343 4443 4E45          #  #CDCNE
1180EC  5423 004E 904F EF00 1E4E 5E4E 754E 804C   T# N O   N^NuN  L
1180FC  0000 0000 0046 6570 7462 4245 4749 4E5F        FeptbBEGIN_
```

DISPLAY_NETWORK_STATUS (DISNS)

Purpose Displays the status of network solutions connected to the DI. The command returns status for specific network solutions, or, if you do not specify names, status of all network solutions connected to the DI. Network status displays are described following the command examples.

Format **DISPLAY_NETWORK_STATUS**
NETWORK_NAME = list 1..15 of name

Parameters *NETWORK_NAME (NN)*

Logical name of a network solution. You may specify one name or a list of names. If you do not specify this parameter, status of all network solutions connected to the DI is displayed.

Responses Network Status
(Followed by the network status display. See example.)
Within the status display, the following response is inserted if a network solution name is not defined in the DI's logical configuration.
Network name <name> not defined.
No network solutions defined. No network status to report.

Examples

```
senc c='display_network_status',s=mdi_8a

Network Status
network_name = ESCI_NET
network_type = ESCI
network_id = 00000001(16)
network_status = active
network_cost = 000A(16)
device_name = $ESCI6
```

Network Status Display Description

The network status display includes the following information.

network_name
: The logical name of network solution.

network_type
: Type of network solution, such as ESCI (Ethernet).

network_id
: Network ID for the network solution.

network_status
: Operational status of the network. Possible values for status include the following:

  configured
  : Network solution is defined, but not in use by the communications system.

  enabled
  : Network solution is in use by the CDCNET communications system.

  active
  : Network solution is active, and communications are being carried over the network. Link and network protocols are being exchanged.

  congested
  : Network solution is active, but the depth of the transmit queue (the number of messages being sent from the DI) is greater than the congestion threshold established on the configuration command that configured the network solution (see the CONGESTED_THRESHOLD parameter on the DEFINE_ETHER_NET command for more information on the congestion threshold concept).

  loading remote
  : Network solution is being used exclusively to load a DI system connected through the network.

cost
: The routing cost assigned to the network solution. This is a relative measure that is determined from a routing algorithm created and maintained by the Routing Management Entity in the DI. Cost may change depending upon the amount of traffic on the network solution, the state of the network solution, and other factors.

device_name
: The physical name of the interface board in the DI to which the network solution is connected. For Ethernet networks, the board type is ESCI. For channel networks, the board type is MCI. For X.25 networks, the board type is CIM.

DISPLAY_OPERATOR_SUPPORT (DISOS) (NOS Only)

Purpose Displays the user names of the operators currently logged into the NOS host and into NETOU. The display also indicates the status of the connection to the host through the trunk. If the status is DOWN, the connection is down. If the status is ACTIVE, the connection is up and in use.

Format **DISPLAY_OPERATOR_SUPPORT**
TRUNK_NAME = name

Parameters *TRUNK_NAME (TN)*

Displays the trunk name of the logical link which supports the operator connection. If this parameter is not specified, the default trunk is used. The default trunk name is specified by a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk. If the MDI/MTI was not loaded over a channel and no default channel trunk is specified on a DEFINE_SYSTEM command, then no default channel trunk exists.

Responses Operators supported for trunk <name>. Connection is <state>. (Followed by the DISOS display (see example). Within the status display, the following responses will replace the display response if operator support is not defined for a specified trunk or if no operator support is defined for the system).

Operator Support is not defined for trunk <name>.

Operator Support is not defined for the system.

Examples

```
senc c='display_operator_support',s=mdi2

Operators supported for c170_trunk 03.  Connection is active.
oper1
oper2
oper3
  .
  .
opern
```

DISPLAY_PASSTHROUGH_STATUS (DISPS)

Purpose Returns the configuration for the Interactive Passthrough Gateway (IPG) and displays the status of all connections the IPG supports.

Format **DISPLAY_PASSTHROUGH_STATUS**

Parameters None.

Responses The response provides five lines of header information about the passthrough service, followed by a 4-line entry for each unique server connected to the IPG.

The header information includes the configured passthrough service title, the status of the passthrough service, the value of the default inactivity timer and the total number of server and client connections. The total number of server connections indicates the number of connections the site has configured to connect to this instance of the IPG. The total number of client connections indicates the number of server connections in actual use.

Each server entry identifies a unique server title connected to the IPG. Each server entry includes the server title, the inactivity timer for this server, the number of server connections represented by the server title and the number of clients currently connected to server connections with this server title.

The format of the header and server entries follows. One or more server entries may follow a header entry.

Service Title	:<DEFPS title>
Status	:<started/stopped>
Default Inactivity Timer	:<DEFPS inactivity timer value>
Total Server Connections	:<Total connected Servers>
Total Client Connections	:<Total connected Clients>

Server Title	:<DEFPT title
Inactivity Timer	:<DEFPT inactivity timer value>
Server Connections	:<Servers with this title>
Client Connections	:<Clients accessing this title>

See example of successful status display.

--ERROR-- Passthrough Service not defined.

Examples

```
senc c='display_passthrough_status'

Default Inactivity Timer        600
Status                          STARTED
Service Title                   PASSTHROUGH
Total Server Connections        32
Total Client Connections        15

Server Title                    ARHBE
Inactivity Timer                300
Server Connections              32
Client Connections              4

Server Title                    ARHBE1
Inactivity Timer                300
Server Connections              16
Client Connections              6

Server Title                    ARHBE2
Inactivity Timer                600
Server Connections              16
Client Connections              5
```

DISPLAY_RECORDER_LOG_GROUP (DISRLG) (NOS Only)

Purpose

Displays the log groups supported by an Independent Log Management Entity (ME) at a NOS MDI or MTI, and the priority of the ME for each log group. For this release of CDCNET, only one log group can be defined per channel trunk for each NOS MDI or MTI. The display also indicates the state of the Independent Log ME connection to the host through the channel trunk. The state may be:

`down` Connection is down.

`active` Connection is up and in use.

Format

DISPLAY_RECORDER_LOG_GROUP
TRUNK_NAME = *name*

Parameters

TRUNK_NAME (TN)

The trunk name of the logical link for which the defined recorder log group is to be displayed. If trunk name is not specified, the recorder log groups defined for all trunks are displayed.

Responses

Recorder Log Groups.
(Followed by the display of log groups. See example. If no recorder log groups are defined for the DI, or if log groups are not defined for a specified trunk, the following responses replace the log group display).

Recorder log groups are not defined for trunk <name>.

Recorder log groups are not defined for the system.

Examples

```
senc c='display_recorder_log_group',s=mdi3

Recorder Log Groups for trunk $mci4.  Connection is active.
Log Group                            Priority
CATENET                              1
```

DISPLAY_REMOTE_LOAD_SUPPORT (DISRLS)

Purpose Returns the current configuration and status of the remote load support of a system. The returned configuration information includes the values of all parameters supported by the DEFINE_REMOTE_LOAD_SUPPORT command. The returned status information includes the status of the load help provided by the remote systems.

Format **DISPLAY_REMOTE_LOAD_SUPPORT**

Responses Remote Load Support Configuration
(Followed by the display. See examples).
The following response replaces the display if remote load support is not defined for the system.
Remote Load Support is not defined for this system.

Examples In this example, the Initialization Management Entity is loading or dumping a remote system when this command was received.

```
senc c='display_remote_load_support'

Remote Load Support Configuration

priority = 3                  concurrent load limit = 4
restricted networks           HDLC_1
                              HDLC_2

Remote Load Support Status

network name             remote system id        load status

ETHERNET_1               0800250A1312            dumping
                         0800250A1313            loading

ETHERNET_2               0800250A1314            loading
```

In this example, the Initialization Management Entity is not loading or dumping any system when the command was received.

```
senc c='display_remote_load_support'

Remote Load Support Configuration

priority = 3                  concurrent load limit = 4
restricted networks           HDLC_1
                              HDLC_2


Remote Load Support Status

At present no remote system is being loaded or dumped.
```

DISPLAY_ROUTING_STATUS (DISRS)

Purpose Displays the operating status of the Routing Management Entity (ME) in a DI. Information displayed includes the network IDs known to the Routing ME (IDs stored in the routing table) and the relay count (number of hops) to the known Network IDs.

Format **DISPLAY_ROUTING_STATUS**

Responses Routing M-E Status
Followed by the status display. (See example).
If no Routing table is defined (no locally attached networks are defined), the following response replaces the status display).

No routing table is defined for this system.

Remarks For more information on Routing Management Entity, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
senc c='display_routing_status',s=tdi3

Routing M-E Status
          Network ID       Relay Count
          00000051          Directly Connected Network
          00000052          2
          00000053          1
```

DISPLAY_SERVICE_DISPLAY (DISSD)

Purpose Returns a list of interactive service names and their associated status text included in the service availability display that is shown when a terminal user enters the DISPLAY_SERVICES terminal user command.

Format **DISPLAY_SERVICE_DISPLAY**
SERVICES = list 1..16 of name or keyword value
DISPLAY_OPTION = keyword value

Parameters *SERVICES or SERVICE (S)*

Lists the interactive service names that are to be compared with the list of displayable services in the DISPLAY_SERVICES command for the terminal. The keyword value ALL specifies all service names.

DISPLAY_OPTION (DO)

Specifies the detail level of information returned about the displayed services. The following keyword values are allowed.

Keyword Value	Description
SUMMARY (S)	Selects only the service names for display.
EXPANDED (E)	Selects the service names and accompanying text for display.

Default is SUMMARY

Responses Services in the displayable list.
Followed by a list of service_name, text, down_text, and temporary_down_text.

<Service_name>

T: <text>
DT: <down_text>
TDT: <temporary_down_text>

Services not in the displayable list.

(Followed by the services you listed on the command, which were not in the displayable services list.)

<Service_name>

--ERROR-- No services defined in displayable list.

Remarks Also see the CHANGE_SERVICE_DISPLAY and CHANGE_SERVICE_DISPLAY_TEXT commands later in this chapter.

Examples

```
senc c='display_service_display'

Services in the displayable list.

ARH817
T:   Call x2830 for information about this service.
DT:  CEs on the machine until the problem is fixed.
     No Estimate at this time.
TDT: Scheduled down time from 8:00 - 9:00A.M.

ARH907
TDT: 907 will go down at 8:00 tonight for maintenance.
```

DISPLAY_SOFTWARE_LOAD_STATUS (DISSLS)

Purpose Returns the status of the software modules loaded in a DI. Lists modules as either retained or not retained. Retained modules remain loaded when not in use.

Format **DISPLAY_SOFTWARE_LOAD_STATUS**

Responses Software Load Status.
(Followed by the load status display. See example.)

Examples senc c='display_software_load_status',s=system_4

```
Software Load Status
name                              retained
-------------------------------------------
xerox_transport                   yes
intranet                          yes
internet                          yes
routing                           yes
hardware_scp                      no
  :
```

Software Load Status Description

The software load status display describes each software component as follows.

name Name of software module.

retained Indicates whether software module is loaded and kept in the DI.

Yes Means the module is retained in the DI and is not unloaded, even if it is not being used.

No Means the module will be made available for unloading when it is no longer being used; the retain flag for the module is not set and the module is still in use.

DISPLAY_SOURCE_ALARMS (DISSA)

Purpose Displays the list of alarms to be sent to your network operations station from a DI and the alarm groups supported by the Log Support Application in the DI.

Format **DISPLAY_SOURCE_ALARMS**

Responses

Source Alarms

Alarm groups:
<list of alarm groups>
Alarm message numbers defined:
<list of alarm message numbers>

(If no source alarm groups are defined for the system, the following response will replace the source alarm group display):

No source alarm groups defined.

(If no source alarm messages are defined, the following response will replace the list of alarm message numbers):

No alarm message numbers defined.

Examples

```
senc c='display_source_alarms',s=tdi1

                    Source alarms
Alarm groups:
   CATENET
Alarm message numbers defined:
   17..82       198       252..280
```

DISPLAY_SOURCE_LOG_GROUP (DISSLG)

Purpose Displays the log groups to which the DI belongs and the messages to be logged for each group. The messages for each log group comprise the log messages reported by a DI.

Format **DISPLAY_SOURCE_LOG_GROUP**

Responses

Source Log Groups

Log Group <name>
Log message numbers defined:
<list of message numbers>

(A response is returned for each log group that is defined. If no messages are enabled for a log group, the following response replaces the message list for the group):

No log message numbers defined.

(If no source log groups are defined for the DI, the following response replaces the log group display):

No source log groups defined.

Examples

```
senc c='display_source_log_group',s=tdi_78

              Source Log Groups
Log Group CATENET
Log message numbers defined:
1..200          300          350
```

DISPLAY_SYSTEM_OPTIONS (DISSO)

Purpose Displays the current value of DI system program attributes. These attributes include memory management parameters and system function options. This command can be used to determine the location of the network's master clock for DIs supported by NOS hosts.

Format **DISPLAY_SYSTEM_OPTIONS**
DISPLAY_OPTION = list of keyword value

Parameters *DISPLAY_OPTION (DO)*

Specifies one or more of the system attributes for display. These attributes are defined by parameters on the DEFINE_SYSTEM command that configured this DI. For more information on these parameters, refer to the DEFINE_SYSTEM command description. Allowed keyword values include the following.

Keyword Value	Description
SYSTEM_NAME (SN)	The DI's title as it appears in the CDCNET directory (minus the $SYSTEM_ prefix).
DATA_BUFFER_SIZE (DBS)	Size, in bytes, of the system data buffers.
BUFFER_PERCENTAGE (BP)	Percentage of total SMM memory allocated to system buffers.
BUFFER_BOUNDARY_ PERCENTAGES (BBP)	Percentages of available buffers corresponding to boundaries between the four levels of DI buffer availability. The first percentage specifies the boundary between GOOD and FAIR buffer availability. The second percentage specifies the boundary between FAIR and POOR buffer availability. The third percentage specifies the boundary between POOR and CONGESTED.
MEMORY_BOUNDARY_ PERCENTAGES (MBP)	Percentages of available memory that correspond to boundaries between the four levels of DI memory availability. The first percentage specifies the boundary between GOOD and FAIR buffer availability. The second percentage specifies the boundary between FAIR and POOR availability. The third percentage specifies the boundary between POOR, and CONGESTED).

Keyword Value	Description
MEMORY_MANAGER_PERIOD (MMP)	Interval, in seconds, that the DI memory manager executes to maintain the DI buffer and memory state.
RESERVED_SYSTEM_SPACE (RSS)	Number of bytes reserved in the free memory pool for executive internal allocations.
STANDARD_STACK_SIZE (SSS)	Size, in bytes, of the task's stack size when the initiator of the task does not specify a stack size to the executive.
DEFAULT_CHANNEL_TRUNK (DCT)	The value for this display option is determined by the DI load process and is not used for this release of CDCNET. The channel trunk field is displayed in the system options display, but use of the CHANNEL_TRUNK parameter in commands is not recommended.
ROUTING_SYSTEM (RS)	Identifies a system that distributes routing information data units when the parameter value is TRUE. Parameter default value is FALSE. CDCNET release 1.2 does not support communities, thus the routing_system parameter value is always FALSE, and never displayed.
CLOCKING_SYSTEM (CS)	Indicates whether or not this DI is the master clock. For NOS-connected DIs, there must be at least one such DI identified in the catenet.
ALL	Selects display of all options.

Default is ALL.

Responses System options.
(Followed by the options display. See example.)

Remarks For more information on these parameters, refer to the DEFINE_SYSTEM command description in this manual.

Examples

```
senc c='display_system_options',s=north_tdi_1

System options
system_name = engineering_tdi_1
data_buffer_size (before reset) = 160
data_buffer_size (after reset) = 160
buffer_percentage = 75
buffer_boundary_percentages = (40,20,5)
memory_boundary_percentages = (40,15,2)
memory_manager_period = 20
reserved_system_space (before reset) = 16384
reserved_system_space (after reset) = 16384
standard_stack_size = 16384
default_channel_trunk = default channel trunk unknown
clocking_system = no
```

DISPLAY_TEST_STATUS (DISTS)

Purpose

Allows you to monitor the progress of an online diagnostics test or display the completion status of an onboard or online diagnostics. The command response indicates the current status of online diagnostics in progress or the completion status of the last onboard, online, or inline diagnostics that was executed on the specified device. Use this command to get the results of online, onboard, and inline diagnostics. For online diagnostics, send this command after you receive a response to the command that starts the diagnostic test. The fields in the test status display are described at the end of this command description.

Format

DISPLAY_TEST_STATUS
DEVICE_NAME = name

Parameters

DEVICE_NAME (DN)

Physical name of the hardware device, consisting of the $ character, a board type and board slot number. Example physical names include $MCI2, $CIM3, $LIM1, $LIM2_PORT1, $ESCI4, and $URI3.

Responses

<device_type> test status.

<device slot number information> (see example).

<device diagnostic status information> (see example).

--ERROR-- Device <device_name> not installed in system.

Examples

The following example shows a status display for an online test that is currently running.

```
senc c='display_test_status device_name=$lim1_port2',s=tdi2

PORT test status
CIM slot number = 3
LIM slot number = 1
PORT number = 2
RUNNING online version 0901
Testing internal loopback
pass count = 50  total errors = 3
```

The following example shows a status display for online and onboard tests that failed.

```
senc c='display_test_status device_name=$cim5',s=tdi2

CIM test status
CIM slot number = 5
FAILED on-line version 10H1 01/24/86 14.43.31
Testing CIM/SMM interface
pass count = 5   error code = 1230  total errors = 1


CIM test status
CIM slot number = 1
FAILED on-board version 09A1 01/24/86 14.43.31
```

Displays for device-passing tests:

```
senc c='display_test_status device_name=$mci1,s=mdi2

MCI test status
MCI slot number = 1
PASSED on-board version 08H1 01/16/86 14.34.21
```

In this example, a CIM online test passed on 01/16/86, a LIM online test failed on 03/18/86, a port online test failed 04/28/86, and a URI online test failed on 04/29/86. This example illustrates the use of a URI/LIM/PORT failure summary on the CIM test's PASSED response to indicate the actual status of the CIM and its URIs, LIMs and ports.

```
senc c='display_test_status device_name=$cim1',s=tdi2

CIM test status
CIM slot number = 1
PASSED on-line version 10H1 01/16/86 14.34.21
pass count = 10
  LIM/PORT failure summary:
    FAILED lim 4        on-line  version 10H1 03/18/86 04.18.01
    FAILED lim 5 port 1 on-line  version 10H1 04/28/86 10.21.22
    FAILED uri 6        on-line  version 2301 04/29/86 07.55.20
```

In this example, a LIM onboard test passed on 01/16/86, and a port online test failed on 01/24/86 and 01/25/86. This example illustrates the use of a PORT failure summary on the LIM test's PASSED response to indicate the actual status of the LIM and its ports.

```
senc c='display_test_status device_name=$lim3',s=tdi2

LIM test status
CIM slot number = 6
LIM slot number = 3
PASSED on-board version 10H1 01/16/86 14.34.21
  PORT failure summary:
    FAILED lim 3 port 1 on-line  version 10H1 01/24/86 14.43.30
    FAILED lim 3 port 3 on-line  version 10H1 01/25/86 10.21.22
```

Diagnostic Test Status Display Description

The test status display includes the following information (as appropriate for the device being tested).

Slot numbers for the device:

Slot number for large board devices (such as CIM and ESCI).

For LIM or URI devices, the CIM slot number as well as the LIM or URI slot number.

For PORT devices, the CIM and LIM slot numbers as well as the PORT number.

Status of last test on device, including the following:

Test status (RUNNING, PASSED, FAILED, or NO TEST HAS RUN ON DEVICE <device_name> SINCE DI WAS LAST RESET).

Test type (ON-BOARD, ONLINE, or INLINE).

Version number of test.

For completed tests that PASSED:

Date and time of test.

Pass count (only for on-line tests)

For online diagnostics tests:

Pass count.

A summary of the failed tests on device subassemblies, such as the status of failed URI, LIM or port tests for CIM tests. If no subassembly tests have failed, NO ERRORS FOUND is reported for the summary.

For completed tests that FAILED:

Date and time of test.

For online diagnostics tests:

Failed operation (the area in which test failed).

Pass count.

Error code of first failure. Note that error code values are undocumented. Refer to the appropriate command response in the CDCNET Diagnostic Messages manual.

Total errors.

For devices for which no test has run since the last DI reset (device is not tested by onboard diagnostics):

Date and time of last DI reset.

For inline diagnostics tests:

Error code of last failure.

Total errors.

A summary of the failed tests on subassemblies to a board, such as the status of failed LIM, URI or port tests for CIM tests. If no tests on subassemblies have failed, the message NO ERRORS FOUND is reported for the summary.

For RUNNING online tests:

Current operation (the area that is being tested).

Pass count.

Total errors.

DISPLAY_XNS_TRANSPORT_STATUS (DISXTS)

Purpose

Displays the operating status of XNS (Xerox Networking System) Transport, the status of the specific service access points (SAPs) serviced by XNS Transport and the status of specific connections serviced by XNS Transport. SAPs and connections are specified by their SAP IDs and connection IDs (these numbers are displayed in expanded and SAP XNS Transport status displays). If both SAP and connection IDs are entered, then the command displays status for all entered SAPs and all entered connection IDs.

If no parameters are entered for a DISPLAY_XNS_TRANSPORT_STATUS command, the command returns the summary status display. XNS Transport status displays are described following the command examples.

Format

DISPLAY_XNS_TRANSPORT_STATUS
SAP = list 1..15 of 0..0FFFFFF(16)
CONNECTION = list 1..15 of 0..0FFFFFF(16)
DISPLAY_OPTION = list of keyword value

Parameters

SAP (S)

Specifies the ID of the SAP to display.

CONNECTION (C)

Specifies the ID of the connection to display.

DISPLAY_OPTION (DO)

Selects the level of status for the general status, SAP status or connection status display. The following keyword values are allowed.

Keyword Value	Description
SUMMARY (S)	Selects the summary display.
EXPANDED (E)	Adds the expanded information to the display.

Default is SUMMARY.

Responses

XNS transport Status.
(Followed by the status display. See example. Within the status display, the following responses will be inserted if a SAP ID or connection ID is unknown, or if a SAP has no connections established).

SAP <0..0FFFFFF(16)> unknown.

Connection <0..0FFFFFF(16)> unknown.

No connections established for this SAP.

Remarks

For more information on Xerox Transport, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

This example shows a summary XNS transport status display.

```
senc c='display_xns_transport_status',s=tdi5

XNS Transport Status
number of SAPs = 2
number of connections = 8
number transport congested = 0
number user congested = 0
number incoming connections = 4
number outgoing connections = 4
```

This example shows an expanded XNS Transport status display.

```
senc c='display_xns_transport_status do=e',s=tdi5

XNS Transport Status
number of SAPs = 2
number of connections = 8
number transport congested = 0
number user congested = 0
number incoming connections = 4
number outgoing connections = 4

SAP id    SAP table   number of   transport  user
          address     connections congested  congested
1046(16)  104BC6(16)       2          0          0
1B3C(16)  104ACA(16)       2          1          0
```

This example shows status display for specific SAP IDs.

```
senc c='display_xns_transport_status ..
sap=(1046(16),1B3C(16))',s=tdi5

XNS Transport Status

SAP id 1046(16)

  connection id 47AC(16)      connection table address = 110C28(16)
  peer internet SAP   = 000000020800252A022313FE
  peer connection id = 1238(16)

SAP id 1B36(16)

  connection id = 1114(16)     connection table address = 114028(16)
  peer internet SAP   = 000000020800252A022303FF
  peer connection id = 1236(16)
```

This example shows status display for a specific connection.

```
senc c='display_xns_transport_status connection=47ac(16)',s=tdi5

XNS Transport Status
connection id = 47AC(16)
connection state = open
connection table address = 110C28(16)
service SAP id = 1046(16)             connection SAP id = 5225(16)
average round trip is 375 ms          length data receive queue = 0
length normal data queue = 0          length normal ack queue = 1
length expedited data queue = 0       length expedited ack queue = 0
connection is not user congested
connection is not transport congested
```

XNS Transport Status Display Description

The summary XNS Transport status display includes the following information about XNS Transport and connections.

`number of SAPs`	The number of service access points (SAPs) serviced by XNS Transport.
`number of connections`	The number of specific connections serviced by XNS Transport.
`number transport congested`	The number of connections that are transport-congested (the transport window has closed).
`number user congested`	Number of connections that are user-congested (the connection user has requested data indications to stop).
`number incoming connections`	The number of connections coming into XNS Transport.
`number of outgoing connections`	The number of connections going out of XNS Transport.

The expanded display adds the following information (see examples for fields being described).

`SAP id`	ID numbers of all SAPs.
`SAP table address`	Memory addresses of all SAPs.
`number of connections`	The number of connections in each SAP.
`transport congested`	The number of connections for the SAP that are transport-congested (the transport window has closed).
`user congested`	The number of connections for the SAP that are user-congested (the user has requested message indications to stop).

If SAP IDs are specified using the SAP parameter, the command returns the following information per SAP.

`connection id`	Connection IDs for all connections on this SAP.
`connection table address`	Memory address of each connection.
`peer internet SAP`	Peer Internet address for each connection; the memory address of the Internet SAP at the other end of the connection.
`peer connection id`	Peer connection ID for each connection; the ID assigned to the connection at the other end of the connection.

Connection IDs are unique for all connections within a given Service SAP. They are not unique for all connections within a DI. The display of one connection ID may result in a display of many connections across many Service SAPs. Connections established within the same millisecond for different SAPs will have the same connection ID.

If connection numbers are entered, the command returns the following information per connection.

connection id	The connection ID specified on the DISXTS command.
connection state	State of the connection, which may be one of the following:
	closed — The connection is disconnected and logically closed.
	connection request sent — A connection request has been sent for the connection.
	connection request received — A connection indication has been received for the connection; awaiting accept or reject from the user.
	confirm acknowledge wait — The user has accepted the connection. Connect confirm PDU sent. Awaiting acknowledgement of the connect confirm.
	open — The connection is open and ready to transmit and receive messages.
	connection timeout wait — No more activity on the connection. Awaiting connection ID timeout to expire.
	disconnecting — The connection is being disconnected.
	unknown — The connection state is unknown. The connection is in an unusable state.
connection table address	Address of each connection.
service SAP id	The SAP identifier for each service.
connection SAP id	If this is the same as the Service SAP ID then this is an outgoing connection; otherwise, this is an incoming connection.
average round trip	Average time in milliseconds required for the process of sending data and receiving the corresponding acknowledgement.
length data receive queue	Length of the normal data receive queue in packets.
length normal data queue	Length of the normal data queue in packets.
length normal ack queue	Length of the normal acknowledge queue in packets.
length expedited data queue	Length of the expedited data queue in packets.

`length expedited data queue`	Length of the expedited acknowledge queue in packets.
`connection is/is not user congested`	Indicates whether the connection is user congested.
`connection is/is not transport congested`	Indicates whether the connection is transport congested.

DISPLAY_X25_GW_OUTCALL_OPTIONS (DISXGOO)

Purpose Displays the outcall titles of an X.25 transparent gateway.

Format **DISPLAY_X25_GW_OUTCALL_OPTIONS**
GATEWAY_NAME = list of 1..15 of name

Parameters *GATEWAY_NAME (GN)*

Specifies the names of the X.25 gateways to display that provide access to remote applications. Default is ALL.

Responses X.25 Gateway outcall configuration
(Followed by the outcall options display. See example for the specified gateway names.)

--ERROR-- X.25 gateway <name> is not defined.

--ERROR-- An X.25 gateway is not defined.

Examples

```
senc c='display_x25_gw_outcall_options'

gateway_name: Telenet

remote_dte_address:  protocol_id:    local_dte_address    title:
3401                 c2              NONE                 PTFS$FOREIGN
facilities:    user_data:
NONE           NONE

remote_dte_address:  protocol_id:    local_dte_address    title:
4801                 c2              2502                 PTFS$REMOTE
facilities:    user_data:
NONE           NONE
```

DISPLAY_X25_NET_OPTIONS (DISXNO)

Purpose Displays the current configuration of an X.25 network.

Format **DISPLAY_X25_NET_OPTIONS**
NETWORK_NAME = list of 1..15 of name
DISPLAY_OPTION = list of keyword value

Parameters **NETWORK_NAME (NN)**

Specifies a list of logical names, of one or more X.25 networks, that were assigned with a DEFINE_X25_NET command.

DISPLAY_OPTION (DO)

Selects one or more of the X.25 network attributes for display. The following keyword values are allowed.

Keyword Value	Description
TRUNK_NAME (TN)	Selects the trunk(s) to be displayed.
REMOTE_DTE_ ADDRESS (RDA)	Selects the remote DTE address for the specified trunk(s).
NETWORK_ID (NI)	Selects the ID of the displayed network.
NETWORK_NAME (NN)	Selects the network to be displayed.
COST (C)	Selects the cost of the selected network to be displayed.
RELAY_ALLOWED (RA)	Displays whether the network allows relays.
ROUTING_INFO_ NETWORK (RIN)	Displays whether the network provides routing information.
NETWORK_PROTOCOL_ ID (NPI)	Displays the protocol identifier of the selected network.
ACCEPT_PDN_ CHARGES (APC)	Displays whether the network accepts cost charges from a public data network.

Responses X.25 Network options
(Followed by the requested list of network options. See examples.)
Within the list of options, the following responses will be inserted if a network name is not defined, is not an X.25 network or if no X.25 networks are defined.

Network <name> is not defined.

Network <name> is not an X.25 network.

No X.25 networks are defined for this system.

Examples

```
senc c='display_x25_net_options nn=x25tymnet'

X.25 Network options
trunk_name = TYMNET_TRUNK_LINE_1
remote_dte_address = 6124825000
network_id = 123456(16)
network_name = X25_TYMNET
cost = 200
relay_allowed = no
routing_info_network = yes
network_protocol_id = C3(16)
accept_pdn_charges = yes
```

START Commands

START_CIM_TEST (STACT)

Purpose

Starts an online diagnostics test for a Communications Interface Module (CIM), all its connected URIs and LIM boards, and their ports.

The CIM diagnostic test should be used only if there are problems on more than one LIM, since all line users must be disconnected and lines must be stopped to run the CIM diagnostic. If problems seem to be confined to one LIM, the LIM test should be run (see START_LIM_TEST command), and if no errors occur while running the LIM test, the Port test should be run (see START_PORT_TEST command).

Format

START_CIM_TEST
DEVICE_NAME = name
REPEAT_PASS = keyword value or integer
SUCCESS_STATE = keyword value
LOGGING = boolean
STOP_ON_ERROR = boolean

Parameters

DEVICE_NAME (DN)

The physical name of the CIM being tested. This name consists of a dollar sign $, the board type (CIM), and the board slot number (0..7), as in $CIM3 for a CIM board in slot 3.

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat (pass). The parameter value may be any integer. The keyword value 0 (zero) specifies that the test will run continuously until you stop the test by a STOP_CIM_TEST command. Default is 1 (one).

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error will cause the test to terminate the current pass and restart testing at the beginning of the next pass.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the diagnostic messages logged in a log file. There are two possible values for this parameter: ON and OFF. ON specifies that diagnostic messages are logged in the log file. OFF specifies that diagnostic messages are not logged. Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether or not you want the test to end if an error condition is encountered. There are two possible values for this parameter: ON and OFF. ON specifies that the test is stopped if any error occurs. OFF specifies that the test is not stopped if any error occurs. See note with the REPEAT_PASS parameter. Default is ON.

Responses

CIM test started, version <version_number>.
CIM slot number = <cim slot number>.

--WARNING-- Device <device_name> test already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--FATAL-- CIM test aborted, version <version_number>.

CIM slot number= <cim slot number>
Unable to start test task.

--FATAL-- CIM test aborted, <version number>.

CIM slot number= <cim slot number>.
Test task stop flag set.

Remarks

In order for this test to run, the device state must be DOWN. Use the CHANGE_ELEMENT_STATE command to change the state of the device.

To get the results of the CIM test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

If you start the CIM test, and the CIM test runs without failure, you do not also have to start the LIM test using START_LIM_TEST. However, you should still run the port test (using START_PORT_TEST), using the EXTERNAL and MODEM loop mode options, to check for problems outside of the CIM and LIM, such as communication line and modem problems.

You can best test LIM select logic failures by running multiple port tests concurrently, using the START_PORT_TEST command. Running the CIM or LIM tests only tests the ports sequentially.

Examples

This example starts an online diagnostics test for a CIM and all its LIMs, running one pass of the test and stopping on the first occurrence of an error.

```
send_command c='start_cim_test device_name=$cim5',s=tdi5

CIM test started, version 0901
CIM slot number = 5
```

This example starts an online diagnostics test for a CIM and all its LIMs. The test will run continuously without stopping for errors. However, since logging is on, any errors encountered during the test will be logged.

```
send_command c='start_cim_test dn=$cim5,rp=0,soe=off',s=tdi5

CIM test started, version 0901
CIM slot number = 5
```

START_ESCI_TEST (STAET)

Purpose

Starts the online diagnostics test on an ESCI board. The ESCI diagnostic test can be used to isolate possible failures on an ESCI board or Ethernet transceivers.

An online diagnostics test affects only the board being tested. Operations and communications traffic for other boards or ports are unaffected. However, during a test the board or port is not available for normal communications traffic. This means that you may not execute online diagnostics on the only board or port supporting the network solution over which the DI receives operations commands from you. This restriction is enforced through the STOP_NETWORK command; since communications must be stopped on the device being tested before the diagnostics can be executed.

Format

START_ESCI_TEST
  DEVICE_NAME = name
  REPEAT_PASS = keyword value or integer
  SUCCESS_STATE = keyword value
  LOGGING = boolean
  STOP_ON_ERROR = boolean

Parameters

DEVICE_NAME (DN)

The physical name of the ESCI being tested. This name consists of a dollar sign $, the board type (ESCI), and the board slot number (0..7). For example, $ESCI4 is the physical name for a ESCI board in slot 4. This parameter has no default parameter.

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat. The parameter value may be any integer. Default is 1. The keyword value 0 specifies that the test will run continuously until you stop the test by a STOP_ESCI_TEST command.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error will cause the test to terminate the current pass and restart testing at the beginning of the next pass.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the diagnostic messages logged in a log file. There are two possible values for this parameter: ON and OFF. ON specifies that diagnostic messages are logged in the log file. OFF specifies that diagnostic messages are not logged. Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether or not you want the test to end if an error condition is encountered. There are two possible values for this parameter: ON and OFF. ON specifies that the test is stopped if any error occurs. OFF specifies that the test is not stopped if any error occurs. See note with the REPEAT_PASS parameter. Default is ON.

Responses

ESCI test started, version <version_number>.
ESCI slot number = <esci slot number tested>.

--WARNING-- Device <device_name> test already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--FATAL-- ESCI test aborted, version <version_number>.

ESCI slot number= <esci slot number>
Unable to start test task.

Remarks

In order for this test to run, the device state must be DOWN. Use the CHANGE_ELEMENT_STATE command to change the state of the device.

To get the results of the ESCI test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

If you specify SUCCESS_STATE=DOWN, you must use the CHANGE_ELEMENT_STATE command when the diagnostic completes to put the device in the ON state.

Examples

This example shows an ESCI online diagnostics test being started for an ESCI board in slot 6 of a DI called North_TDI_1. Logging is to be turned off for this test and no errors will be logged.

```
senc c='start_esci_test device_name=$esci6,l=off',s=north_tdi_1

ESCI test started, version 0901
ESCI slot number = 6
```

START_LIM_TEST (STALT)

Purpose Starts an online diagnostics test on a LIM board and its ports.

The LIM diagnostic test should be run if failures are reported on two or more ports on the same LIM. If no errors occur while running the LIM test, the Port diagnostic test should be run (see START_PORT_TEST command). If problems are reported on more than one LIM, the CIM diagnostic test should be run (see START_CIM_TEST).

Format

START_LIM_TEST
DEVICE_NAME = name
REPEAT_PASS = keyword value or integer
SUCCESS_STATE = keyword value
LOGGING = boolean
STOP_ON_ERROR = boolean

Parameters

DEVICE_NAME (DN)

Physical name of LIM device, consisting of a dollar sign $, board type (LIM) and slot number, as in $LIM5 (device name for LIM board in slot 5).

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat. The parameter value may be any integer. Default is 1. The keyword value 0 specifies that the test will run continuously until you stop the test by a STOP_LIM_TEST command.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error will cause the test to terminate the current pass and restart testing at the beginning of the next pass.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the diagnostic messages logged in a log file. There are two possible values for this parameter: ON and OFF. ON specifies that diagnostic messages are logged in the log file. OFF specifies that diagnostic messages are not logged. Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether or not you want the test to end if an error condition is encountered. There are two possible values for this parameter: ON and OFF. ON specifies that the test is stopped if any error occurs. OFF specifies that the test is not stopped if any error occurs. See note with the REPEAT_PASS parameter. Default is ON.

Responses

LIM test started, version <version_number>.
CIM slot number= <CIM slot number>.
LIM slot number= <lim slot number>.

--WARNING-- Device <device_name> test already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--FATAL-- LIM test aborted, version <version_number>.

CIM slot number= <cim slot number>.
Unable to start test task.

--FATAL-- LIM test aborted, version <version_number>.

CIM slot number= <cim slot number>.
LIM slot number= <lim slot number>.

You receive one of the following abort reasons whenever the START_LIM_TEST aborts.

- You receive the following response when the test task started but terminated prematurely.

  Test task stop flag set

- You receive the following response when the LIM test cannot run because all ports on the LIM are turned OFF. Use the CHANGE_STATE_ELEMENT command to change the hardware to the appropriate state. Refer to the CHANGE_STATE_ELEMENT command elsewhere in this chapter.

  State of all ports is "OFF"

- You receive the following response when no ports are supported on the LIM as indicated by the LIM Status Table. This may occur if the LIM on-board tests fail. Use the DISPLAY_TEST_STATUS command to determine the status of on-board tests.

  LIM Status Table indicates no ports supported on lim

- You receive the following response if the LIM specified on the last line of the response is not one of the following supported types.

  4-channel RS232 (xx=08 (16) through 0F (16))
  RS449 (xx=00 (16) through 07 (16))
  V.35 (xx=20 (16) through 27 (16))

  --FATAL-- LIM test aborted, version <version_number>.

  CIM slot number= <cim slot number>.
  LIM slot number= <lim slot number>.
  Test not allowed for LIM type xx.

- As seen in the following response, there is a special case defined for CIM failures that prohibits starting a lower level test such as a LIM or a port test. That is, if the CIM has failed, you will not be able to start a LIM test until you run a CIM test (using the START_CIM_TEST command).

  --FATAL--

  LIM test aborted, version <version number>.
  CIM slot number= <cim slot number>.
  LIM slot number= <lim slot number>.
  Previous CIM failure requires CIM to be tested first.
  ENTER "start_cim_test dn= <device name>".

Remarks

In order for the LIM test to run, the device state must be DOWN. Use the CHANGE_ELEMENT_STATE command to change the state of the device.

To get the results of the LIM test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

If you specify SUCCESS_STATE=DOWN, you must use the CHANGE_ELEMENT_STATE command when the diagnostic completes to put the device in the ON state.

Examples

```
senc c='start_lim_test device_name=$lim5',s=south_tdi

LIM test started, version 10H3.
CIM slot number = 6.
LIM slot number = 5.
```

START_LINE (STAL)

Purpose

Starts communications over a communication line or a URI line. The terminal interface program (TIP) supporting the line must be defined for this command to succeed.

Format

START_LINE
LINE_NAME = name

Parameters

LINE_NAME (LN)

The logical name of the line assigned by the DEFINE_LINE configuration command.

Responses

Line <line_name> started.

--ERROR-- Line <line_name> already started.

--ERROR-- Line <line_name> not defined.

--ERROR-- TIP for line <line_name> not configured.

--FATAL-- Line start-up failed.

Examples

```
send_command c='start_line line_name=line31',s=engin_tdi_3

Line LINE31 started.
```

START_LINE_METRICS (STALM)

Purpose

Starts the collection and optional reporting of statistics for one or more communication or URI lines. If statistics are already started for lines, they are immediately reported and the report period restarted. The line statistics are recorded by the terminal interface program (TIP) supporting the line. The Network Performance Analyzer (NPA) reports that collect line statistics are TERMRP1 and TERMRP2.

Format

START_LINE_METRICS
   LINE_NAME = list 1..15 of name
   REPORT_INTERVAL = 1..86400
   GROUP = list 1..2 of keyword value
   REPORT = boolean

Parameters

LINE_NAME (LN)

The logical names of any communication or URI lines for which statistics are to be collected.

REPORT_INTERVAL (RI)

Statistic reporting interval, specified in seconds. This parameter indicates how often the statistics will be reported. The maximum interval is 24 hours (86,400 seconds).

GROUP (G)

Specifies the type of statistics group requested to be collected: SUMMARY, EXPANDED, or ALL. Default is SUMMARY statistics.

REPORT (R)

Specifies whether or not statistics should be reported by log messages. The messages will be generated and sent to the CDCNET log file according to the interval set by the REPORT_INTERVAL command. Possible values are YES, generate reporting log messages, and NO, do not generate reporting log messages. Default is YES.

Responses In the current CDCNET release, the START_LINE_METRICS command returns a response listing the line metrics successfully started, and the line metrics not started because of errors during command entry.

Line <line_name> <group_name> metrics started.

(One response for each line and group for which metrics was started.) The following line is also output for a metric if the log message number used to report that the message is not enabled for the DI.

Reporting log message <message_number> not enabled.

When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command.

For lines that are not defined or do not support line metrics, the following lines are inserted.

Line <line_name> not defined.

Line <line_name> <group_name> metrics not supported.

--FATAL-- Line <line_name> metrics start-up failed.

--FATAL-- Line <line_name> metrics start-up failed, not enough memory currently exists for required table space.

Remarks For line statistics to be reported, log message number 166 (line statistics) must be enabled. To check whether this message is enabled, use the DISPLAY_SOURCE_LOG_MESSAGES command. If it is not enabled, enable it using the CHANGE_SOURCE_LOG_GROUP command.

Refer to the NPA manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
send_command c='start_line_metrics ln=bld_3_async_22,g=all',s=west_tdi
```

```
Line BLD_3_ASYNC_22 summary metrics started.
Line BLD_3_ASYNC_22 expanded metrics started.
```

START_MCI_INLINE_TEST (STAMIT)

Purpose

Starts the inline diagnostics testing of an MCI board. An inline diagnostics test shares access to the device being tested with non-diagnostic software, while an online diagnostics test has exclusive access to and control of the device being tested.

Format

START_MCI_INLINE_TEST
DEVICE_NAME = name
MESSAGE_COUNT = integer
MESSAGE_LENGTH_OPTION = keyword value
MESSAGE_INTERVAL = integer

Parameters

DEVICE_NAME (DN)

Specifies the physical name of the MCI to be tested.

MESSAGE_COUNT (MS)

Specifies the number of messages to be transmitted and received as part of this inline test. The default value is 100.

MESSAGE_LENGTH_OPTION (MLO)

Length of the test messages to be transmitted as part of the inline test. The following keywords are valid for this parameter.

N1
N2
N3
N4
N5
N10
N500
N1500
SMALL
LARGE
MIXED

The keywords allow a test message to be either a fixed or relative length (in bytes).

Specify one of the fixed keywords when you want all messages transmitted during the test to be the same length. The fixed length keywords and their values are as follows.

Keyword	Value
N1	1 byte
N2	2 bytes
N3	3 bytes
N4	4 bytes
N5	5 bytes
N10	10 bytes
N500	500 bytes
N1500	1500 bytes

Specify a relative keyword when the transmitted message length can be within a certain range. If you select a relative value, the inline test diagnostic determines the test message length. The same size is not used for all messages. The diagnostic software distributes the test messages length within a range you selected.

The relative length keywords and their values are as follows.

Keyword	Value
SMALL	1 through 500 bytes
LARGE	500 through 1500 bytes
MIXED	1 through 1500 bytes

The default value for this parameter is MIXED.

MESSAGE_INTERVAL (MI)

Specifies the time interval between test messages. Specify the value in milliseconds. The diagnostic inline software delays the specified time before transmitting the next test message. A parameter value of 0 means test messages are transmitted as fast as possible. The default is 0.

Responses

MCI in line test, version <version>
started for device <device_name>

--WARNING-- MCI inline test for device <device_name> is already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "ON" status.

--ERROR-- Device <device_name> not a MCI board.

--ERROR-- Channel trunk for device (device_name) is not defined.

--ERROR-- An NP interface or channel network solution for device (device_name is not defined).

--ERROR-- NP interface for device <device_name> is not up.

--ERROR-- Channel network solution for device <device_name> is not up.

--ERROR-- Unable to start the MCI inline test
Not enough memory is available for the required table space.

--ERROR-- Unable to start the MCI inline diagnostics task.

Examples

```
senc c='start_mci_inline_test device_name = $mci7'

MCI in line test, version 2605
started for device $mci7
```

START_MCI_TEST (STAMT)

Purpose Starts the online diagnostic test on an MCI board.

Format **START_MCI_TEST**
DEVICE_NAME = name
REPEAT_PASS = keyword value or integer
SUCCESS_STATE = keyword value
LOGGING = boolean

Parameters **DEVICE_NAME (DN)**

Physical name of the MCI to be tested. The physical name consists of a dollar sign $, board type (MCI), and the slot number, as in $MCI6 (device name for an MCI in slot 6. There is no default value.

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat. The parameter value may be any integer. Default is 1. The keyword value 0 specifies that the test will run continuously until you stop the test by a STOP_MCI_TEST command.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left in upon successful completion of the diagnostic test. Possbile values are ON and DOWN. On specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the diagnostic messages logged in a log file. There are two possible values for this parameter: ON and OFF. ON specifies that diagnostic messages are logged in the log file. OFF specifes that diagnostic messages are not logged. Default is ON.

Responses

MCI test started, version <version number>

MCI slot number= <mci slot number>

--WARNING--

Device <device_name> test already started

--ERROR--

Device <device_name> not installed in system

--ERROR--

Device <device_name> not in "DOWN" state

--ERROR--

Device <name> test already started. Only one MCI test
is allowed to be active at one time.
Stop Active test
or wait for it to complete.

--FATAL--

MCI test aborted, version <version number>
MCI slot number= <mci slot number>
Unable to start test task

--FATAL--

MCI test aborted, version <version number>
MCI slot number= <mci slot number>
Test task stop flag set

Examples

```
senc c=' start_mci_test device_name = $mci7'

MCI test started, version 10H3
MCI slot number= 7
```

START_NETWORK (STAN)

Purpose Starts communications over an X.25, Ethernet, HDLC, or channel network solution, including starting the underlying X.25, Ethernet, HDLC, or channel trunk.

Format **START_NETWORK**
NETWORK_NAME = name

Parameters **NETWORK_NAME (NN)**

The logical name of the network assigned by a define command that configured the network solution.

Responses <Network_type> network <name> started for trunk <trunk_name>.

--WARNING-- The 3A Command Processor has timed-out waiting for response from SSR. Please check network status for completion of request.

--ERROR-- Network <name> already started for trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> down. Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> off. Unable to start network <network_name>.

--ERROR-- Network <name> is not defined.

--FATAL-- Stream Service Error

This response includes one of the following error messages.

The device manager did not accept a function for the ESCI board.

Unable to initialize ESCI board.

HDLC SSR received error when sending command to DVM.

HDLC SSR received error on start port services.

Not enough memory is currently available for required table space.

Unable to open statisics SAP.

Unable to open memory management SAP.

Unable to initialize MCI board.

--FATAL-- Unable to start task <entry_point_name>.

Examples

```
senc c='start_network network_name=plymouth_net_1',s=ptdi1
```

```
ETHERNET Network PLYMOUTH_NET_1 started for trunk
PLYMOUTH_TRUNK_1.
```

START_NETWORK_METRICS (STANM)

Purpose

Starts the collection and optional reporting of statistics for one or more network solutions. If statistics are already started for a network solution, they are immediately reported and the report period restarted. The statistics for a network solution include statistics from the stream service routine (SSR) supporting the network and statistics from the Intranet (3A) layer. The Network Performance Analyzer (NPA) reports that collect network statistics are ETHRRP1 and ETHRRP2 (for Ethernet network solutions), MCISRP1, MCISRP2, and MCISRP3 (for mainframe channel network solutions).

Format

START_NETWORK_METRICS
NETWORK_NAME = list of name
REPORT_INTERVAL = 1..86400
GROUP = list 1..2 of keyword value
REPORT = boolean

Parameters

NETWORK_NAME (NN)

The name or names of one or more network solutions. The names are those defined during configuration. For example, the network solution name for an Ethernet network solution is defined by the DEFINE_ETHER_NET command. For this CDCNET release, the channel network solution is defined during the DI load process and uses a default logical name. The channel trunk name is the NETWORK_NAME for the channel interface to a NOS host. Network and trunk names can be found by using the DISPLAY_LOGICAL_NAMES command.

REPORT_INTERVAL (RI)

Statistic reporting interval, specified in seconds. This parameter indicates how often the statistics are reported. The maximum interval is 24 hours (86,400 seconds).

GROUP (G)

Level of statistics to be collected: SUMMARY, EXPANDED, or ALL. Default is SUMMARY statistics.

REPORT (R)

Specifies whether or not a reporting message should be generated through log messages. Possible values are YES, generate reporting message, and NO, do not generate reporting message. Default is YES.

Responses

In the current CDCNET release, the START_NETWORK_METRICS command returns a response listing the network metrics successfully started, and the network metrics not started because of errors during command entry.

NETWORK <network_name> <group_name> Metrics started.

(One response for each network and group for which metrics was started.) The following line is also output for a metric if the log message number used to report that the message is not enabled for the DI.

Reporting log message <message_number> not enabled.

When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command.

For networks that are not defined or do not support network metrics, the following lines are inserted.

Network <network_name> not defined.

Network <network_name> <group_name> metrics not supported.

--FATAL-- Network <network_name> metrics start-up failed.

--FATAL-- Network <network_name> metrics start-up failed, not enough memory currently exists for required table space.

Remarks

In order for network statistics to be reported, the following log message numbers must be enabled: for Ethernet statistics, message number 639, and for MCI (channel) statistics, message number 562. To check if these messages are enabled, use the DISPLAY_SOURCE_LOG_GROUP command. If these messages are not enabled, enable them using the CHANGE_SOURCE_LOG_GROUP command.

Refer to the NPA manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
senc c='start_network_metrics nn=bld_3_ethernet,g=(summary,expanded)',..
s=mdi_01

Network BLD_3_ETHERNET summary metrics started.
Network BLD_3_ETHERNET expanded metrics started.
```

START_NP_INTERFACE (STANI) (NOS Only)

Purpose

Starts the Network Products (NP) protocol over a NOS mainframe channel to a NOS system and starts the underlying channel trunk protocol if it has not already been started.

Format

START_NP_INTERFACE
INTERFACE_NAME = name

Parameters

INTERFACE_NAME (IN)

The logical name of the interface assigned by the DEFINE_NP_INTERFACE command.

Responses

NP_interface <interface_name> started.

--WARNING-- NP interface <interface_name> command processor has timed-out waiting for a response from the NP interface task.

--ERROR-- NP interface <interface_name> is not defined.

--ERROR-- NP interface <interface_name> already started.

--FATAL-- Unable to start NP interface <interface_name>. Unable to start task SVM.

--FATAL-- Unable to start NP interface <interface_name>. Unable to start task BIP.

--FATAL-- Unable to start NP interface <interface_name>. Unable to send ITM to NP interface task.

--FATAL-- Unable to start NP interface <interface_name>. Memory management sap table not found.

--FATAL-- Not enough memory is currently available for required table space.

--FATAL-- Unable to start NP interface <interface_name>. Unknown status returned from open memory sap.

Examples

```
senc c='start_np_interface in=cyber_109',s=mdi2

NP interface CYBER_109 started.
```

START_PASSTHROUGH_SERVICE (STAPS)

Purpose Starts the interactive passthrough service. The service allows passthrough ports to connect to the Interactive Passthrough Gateway and register their respective titles.

Format **START_PASSTHROUGH_SERVICE**

Responses Passthrough Service started.

--ERROR-- Passthrough Service not defined or already started.

Examples

```
senc c='start_passthrough_service'

Passthrough Service started.
```

START_PORT_TEST (STAPT)

Purpose

Starts an online diagnostics test on an individual LIM port.

This diagnostic test should be run if failures are reported on only one port or on lines associated with multiple LIMs. Multiple port tests should be run at the same time if failures are reported on lines associated with multiple LIMs.

Format

START_PORT_TEST
  DEVICE_NAME = name
  REPEAT_PASS = keyword value or integer
  SUCCESS_STATE = keyword value
  LOGGING = boolean
  STOP_ON_ERROR = boolean
  LOOP_MODE = keyword value
  MODEM_CLASS = 1..6

Parameters

DEVICE_NAME (DN)

Physical name of the device to be tested, consisting of a dollar sign ($), board type (LIM), its slot number, the keyword PORT, and port number. For example, $LIM3_PORT1 is the device name for port 1 on the LIM board in slot 3.

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat. The parameter value may be any integer. Default is 1 (one). The keyword value 0 (zero) specifies that the test will run indefinitely until you stop the test by a STOP_PORT_TEST command.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error will cause the test to terminate the current pass and restart testing at the beginning of the next pass.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the diagnostic messages logged in a log file. There are two possible values for this parameter: ON and OFF. ON specifies that diagnostic messages are logged in the log file. OFF specifies that diagnostic messages are not logged. Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether or not you want the test to end if an error condition is encountered. There are two possible values for this parameter: ON and OFF. ON specifies that the test is stopped if any error occurs. OFF specifies that the test is not stopped if any error occurs. See note with the REPEAT_PASS parameter. Default is ON.

LOOP_MODE (LM)

Selects method of loopback for the LIM port. The following three keyword values, and corresponding loopback modes are allowed.

INTERNAL (I) — Checks the internal logic of the LIM port by sending a signal through it, but not through the board's drivers or receivers. Does not check anything past the LIM port.

EXTERNAL (E) — Checks transmitters and receivers on the LIM port. This loopback mode requires a loopback plug jumper to be placed on the LIM port before running the loopback test.

MODEM (M) — Checks the LIM port including external cables, the modem or modems, and the communication line. The modem (local or remote) must be manually switched to loopback data towards the LIM. Refer to the specific modem user manual to determine the proper switch setting. To run the modem loopback test, specify MODEM when entering the START_PORT_TEST command and select the loopback on the local or remote modem before starting the test.

The use of the external clock is a strap selectable feature on the RS449 Model A LIM. The strap must be removed to run the external loopback test.

Port	Strap Location	Pins
0	63G3	9-12
1	44K6	4-17

The MODEM loopback test can also be used to check the LIM port to terminal connections when modems are not present. This can be done by using a loopback plug at any point in the LIM port to terminal path. The modem loopback test will raise RTS and DTR and check for CTS and DCD to be active and for TxD to be tied to RxD. RS-232-C and RS-449 loopback plugs are included in the Customer Maintenance Kit. Refer to the Parts Data section of the CDCNET Troubleshooting Guide for the correct loopback plug part number. To run the modem loopback test on this type of configuration, specify MODEM when entering the START_PORT_TEST command and ensure that the correct loopback plug is installed. Also, if the LIM port has not yet been configured as an ASYNC line, the MODEM_CLASS parameter must be specified with a value of 2, 4, or 6 (see MODEM_CLASS parameter description).

The following table shows the functional loopback required to run the modem loopback test.

Signal Name	RS232	CCITT	RS449
Transmit Data (TxD)	BA	103	SD
Receive Data (RxD)	BB	104	RD
Request to Send (RTS)	CA	105	RS
Clear to Send (CTS)	CB	106	CS
Data Terminal Ready (DTR)	CD	108/2	TR
Data Carrier Detect (DCD)	CF	109	RR

Default is INTERNAL. Both EXTERNAL and MODEM loopback will first execute INTERNAL loopback testing. Also, EXTERNAL and MODEM loopback methods may only be selected for LIM port testing, not for other board tests. Run the INTERNAL and EXTERNAL options before running the MODEM option.

MODEM_CLASS (MC)

Selects the maximum modem speed for a group of MODEM types. This parameter will only be used when LOOP_MODE=MODEM is selected, and it is required if you choose MODEM loopback, and then is required only if the port has not been configured or has been configured as a line with auto recognition. The following modem class table provides information about modem classes and speeds. There is no default value for this parameter. Refer to the Network Troubleshooting Guide for more information on loopback testing for modems.

Modem Type	Operating Mode	Maximum Speed	Modem Class
Bell 201C	Sync	2,400	1
Bell 103	Async	300	2
Bell 113	Async	300	2
Bell 212A	Sync	1,200	3
Bell 212A	Async	1,200	4
Avanti 2200	Sync	56,000	5
Gandalf LDS260	Sync	56,000	5
Avanti 2200	Async	19,200	6

Responses

PORT test started, version <version_number>.
CIM Slot number = <cim slot number>.
LIM Slot number = <lim slot number>.
PORT number = <port number>.

--WARNING-- Device <device_name> test already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--ERROR-- Modem class (MC) parameter required for modem loopback.

--FATAL-- PORT test aborted, version <version_number>.
CIM slot number= <cim slot number>.
LIM slot number= <lim slot number>.
PORT number= <port number>.
<abort reason - See below*>.

*Unable to start test task

You receive the following response if the LIM is none of the listed supported types.

4-channel RS232 (xx=08 (16) thru 0F (16))

RS449 (xx=00 (16) thru 07 (16))
V.35 (xx=20 (16) thru 27 (16))

*Test not allowed for LIM type xx

You receive the following response when you try a port test on a RS232 LIM with an invalid ID type. Only LIM testing is allowed. The port test is allowed on RS232 LIMs with an id type of 09 thru 0E (16).

*Port test is not allowed for LIM type xx

*ENTER START_LIM_TEST DN= <device_name>

You receive the following response when the test task started but terminated prematurely.

*Test task stop flag set.

You receive the following response after an attempt was made to run the modem loopback test without indicating the modem class. The modem class parameter is required if the line has not been configured or has been configured as an auto-recognition line. Include the modem class parameter or reconfigure the line and reenter the START command to run the test.

*Modem class (MC) parameter is required for modem loopback when line has not been configured or is an auto-recognition line.

Remarks

There is a special case defined for CIM and LIM failures that prohibits starting a lower level test such as a port test. That is, if the CIM or LIM has failed, you will not be able to start a port test until you run a CIM or LIM test (using the START_CIM_TEST or START_LIM_TEST commands). In such a case, if a START_LIM_TEST command is attempted, an abort response will be issued with a reminder to run the higher level test, as in the following example.

```
--FATAL--
PORT test aborted, version 0901
CIM slot number = 6
LIM slot number = 3
Port number = 2
Previous LIM failure requires LIM to be tested first
Enter "START_LIM_TEST DN=$LIM3"
```

In order for the port test to run, the device state must be DOWN. Use the CHANGE_ELEMENT_STATE command to change the state of the device.

To get the results of the port test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

If you specify SUCCESS_STATE=DOWN, you must use the CHANGE_ELEMENT_STATE command when the diagnostic completes to put the device in the ON state.

Examples

```
senc c='start_port_test device_name=$lim3_port1',s=tdi_5

PORT TEST STARTED, VERSION 10H3.
CIM slot number = 5.
LIM Slot number = 3.
Port number = 1.
```

START_PROCESS_METRICS (STAPM)

Purpose

Starts the collection and optional reporting of statistics for the specified software processes and statistic groups. If statistics are already started for the software processes specified, they are immediately reported and the report period restarted. Software statistics are reported in the following Network Performance Analyzer (NPA) reports: DIOSRP1, DIOSRP2, DIOSRP3, and DIOSRP4 (for DI operating system statistics), and SESSRP1 (for Session layer statistics).

Format

START_PROCESS_METRICS
PROCESS = list 1..15 of name
REPORT_INTERVAL = 1..86400
GROUP = list 1..3 of keyword value
REPORT = boolean

Parameters

PROCESS (P)

Logical name of a communications system process. The following software process names are supported.

Intranet
XNS_internet
Generic_transport
XNS_transport
Session
NP_IVT_GW
Routing
File_access
Directory
Log_Support_Application (source log)
Independent_Log_ME (record log)
Command
OSA (Operator Support Application)
LCM (Line Control Module)
System
DOD_Internet
TCP (Transmission Control Protocol)
Telnet_Interface

REPORT_INTERVAL (RI)

Statistic reporting interval, specified in seconds. This parameter indicates how often the statistics will be reported. The maximum interval is 24 hours (86,400 seconds).

GROUP (G)

Type of statistics group requested. Possible keyword values include the following: SUMMARY, EXPANDED, DEBUG, and ALL. Debug statistics provide expanded statistics plus additional statistics about the process being sampled for use by analysts, such as global memory addresses and amount of global memory used. Default is SUMMARY statistics.

REPORT (R)

Specifies whether or not a reporting message should be generated via a log message. Possible values are YES, generate reporting message, and NO, do not generate reporting message. Default is YES.

Responses

<process_name> <group_name> metrics started.

In the current CDCNET release, the START_PROCESS_METRICS command returns a response listing the process metrics successfully started and the process metrics not started because of errors during command entry.

(One response for each process and group for which metrics was started.) The following line is also output for a metric if the log message number used to report that metric is not enabled for the DI.

Reporting log message <message_number> not enabled.

When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command.

For processes that are not defined or do not support process metrics, the following lines are inserted:

This process <process _name> unknown to statistics.

<process _name> <group_name> metrics not supported.

Specified group not suppported for this process.

--FATAL-- <process _name> metrics failed.

--FATAL-- <process _name> metrics failed, not enough memory currently exists for required table space.

Remarks

In order for process statistics to be reported, the following log message numbers must be enabled: for DI operating system statistics, message number 299, and for Session Layer statistics, message number 737. To check if these messages are enabled, use the DISPLAY_SOURCE_LOG_GROUP command. If these messages are not enabled, enable them using the CHANGE_SOURCE_LOG_GROUP command.

Refer to the NPA manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
senc c='start_process_metrics p=xns_transport,..
g=(summary,expanded)',s=mdi_3

XNS_TRANSPORT summary metrics started.
XNS_TRANSPORT expanded metrics started.
```

START_SERVER_TELNET_GW (STASTG)

Purpose

Starts the host terminal gateway service. The gateway accepts TELNET connections from remote users and connects these users to the defined host interactive terminal service.

Format

START_SERVER_TELNET_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the server TELNET host gateway defined by a DEFINE_SERVER_TELNET_GW command.

Responses

Server TELNET gateway <gateway_name> is started.

--ERROR-- Server TELNET gateway <gateway_name> is not defined.

--ERROR-- Server TELNET gateway <gateway_name> is already started.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
senc c='start_server_telnet_gw gateway_name=gw_to_cyber',s=ndi1

Server TELNET gateway GW_TO_CYBER is started.
```

START_TCPIP_GW (STATG)

Purpose

Starts the TCP/IP application interface gateway. The gateway registers titles specified from the DEFINE_TCPIP_GW command to allow host resident applications to make TCP/IP connections.

Format

START_TCPIP_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the TCP/IP application gateway defined by a DEFINE_TCPIP_GW command.

Responses

TCP/IP Gateway <gateway_name> is started to support <protocol> protocol.

--ERROR-- TCP/IP gateway <gateway_name> is not defined.

--ERROR-- TCP/IP gateway <gateway_name> is already defined.

--ERROR-- TCP/IP gateway <gateway_name> is already started.

--FATAL-- TCP/IP gateway <gateway_name> was unable to open SAP.

Examples

```
senc c='start_tcpip_gw gateway_name=ftp_gw',s=ndi1

TCP/IP gateway FTP_GW is started to supported TCP protocol.
```

START_URI_TEST (STAUT)

Purpose Starts the online diagnostics test on an individual unit record interface (URI).

Format

START_URI_TEST
DEVICE_NAME = name
REPEAT_PASS = integer
SUCCESS_STATE = boolean
LOGGING = boolean
STOP_ON_ERROR = boolean
LOOP_MODE = keyword value

Parameters

DEVICE_NAME (DN)

Physical name of the device to be tested, consisting of a dollar sign ($), board type (URI), and its slot number.

REPEAT_PASS (RP)

Specifies how many times you want the test to repeat. The parameter value may be any integer. Default is 1 (one). The parameter value 0 (zero) specifies that the test will run indefinitely until you stop the test by a STOP_URI_TEST command.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error will cause the test to terminate the current pass and restart testing at the beginning of the next pass.

SUCCESS_STATE (SS)

Determines the state in which the hardware device will be left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state will be set to ON if the test completes without error, but remain set to the DOWN state if the test detects an error. DOWN specifies that the state will remain set to DOWN regardless of the test outcome. Default is ON.

LOGGING (L)

Specifies whether you want the messages logged in a log file. This parameter has two possible values: ON and OFF. ON specifies that messages are logged in the log file. OFF specifies that messages are not logged. Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether you want the test to end if an error condition is encountered. This parameter has two possible values: ON and OFF. ON specifies that the test is stopped if any error occurs. OFF specifies that the test is not stopped if any error occurs. See note with the REPEAT_PASS parameter. Default is ON.

LOOP_MODE (LM)

Selects method of loopback for the URI. The following keyword values are allowed.

EXTERNAL (E) External loopback executes internal loopback testing before executing external loopback testing. Install the appropriate loopback plug on the URI board or the printer end of the URI/Printer cable before executing the external loopback tests. Refer to the Parts Data section of the CDCNET Troubleshooting Guide for the correct loopback plug part number.

INTERNAL (I) Internal loopback executes internal loopback testing of the logic of the URI board.

Default is INTERNAL.

Responses

URI test started, version <version number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>

--WARNING-- Device <device_name> test already started.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--FATAL-- URI test aborted, version <version_number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>
Unable to start test task

The following response indicates the test task started but terminated prematurely.

--FATAL--
URI test aborted, version <version_number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>
Test task stop flag set

The following response identifies a CIM failure that will not allow you to start a lower level test such as a URI test. When a CIM fails, you will not be able to start a URI test until you run a CIM test (using the START_CIM_TEST command). When you receive the response, run the START_CIM_TEST command before attempting to run the START_URI_TEST again.

--FATAL--
URI test aborted, version <version number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>
Previous CIM failure requires CIM to be tested first
ENTER "start_cim_test dn=<device_name>

Examples

```
senc c='start_uri_test device_name = $uri5'

URI test started, version 2301
CIM slot number= 6
URI slot number= 5
```

START_USER_TELNET_GW (STAUTG)

Purpose

Starts the user TELNET interactive terminal gateway service. A gateway title or titles are selected with the TITLE (T) parameter of the DEFINE_USER_TELNET_GW command. The gateway title or titles are registered so that CDCNET terminal users can establish TELNET interactive terminal connections with a remote host.

Format

START_USER_TELNET_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the user TELNET gateway defined by a DEFINE_USER_TELNET_GW command.

Responses

User TELNET gateway <gateway_name> is started.

--ERROR-- User TELNET gateway <gateway_name> is not defined.

--ERROR-- User TELNET gateway <gateway_name> is already started.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
senc c='start_user_telnet_gw gateway_name=vax_gw',s=ndi1

User TELNET gateway VAX_GW is started.
```

START_X25_ASYNCTIP (STAXA)

Purpose

Starts the X.25 asynchronous TIP service for the specified X.25 trunks. Allows the X.25 asynchronous TIP to accept terminal connections from the specified trunks. The X.25 trunk must have been previously started.

Format

START_X25_ASYNCTIP
TRUNK_NAME = list 1..32 of name

Parameters

TRUNK_NAME (TN)

The logical name of one or more X.25 trunks for which X.25 asynchronous tip service is to start. Parameter has no default.

Responses

X.25 AsyncTip support started for specified trunks.

--ERROR-- X.25 AsyncTip support not defined for trunk <trunk_name>.

--ERROR-- X.25 AsyncTip support already started on trunk <trunk_name>.

--ERROR-- Duplicate trunk name <trunk_name> specified.

Examples

```
senc c='start_x25_asynctip trunk_name = telenet_2'

X.25 AsyncTip support started for specified trunks.
```

START_X25_GW (STAXG)

Purpose

Starts the specified X.25 gateway and adds title(s) registered for the gateway to the CDCNET directory. START_X25_GW activates any title or titles added with the ADD_X25_GW_OUTCALL command, and reactivates any title or titles previously inactivated (deregistered) when the gateway was stopped. Although the titles were inactivated, they remained known to the gateway. The titles are reactivated when the gateway is restarted.

Format

START_X25_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of an X.25 gateway defined by a DEFINE_X25_GW command.

Responses

X.25 gateway <name> is started.

--ERROR-- X.25 gateway <name> is already started.

--ERROR-- X.25 gateway <gateway_name> is not defined.

Examples

```
senc c='start_x25_gw gn=telenet_gw',s=ndi1

X.25 gateway TELENET_GW is started.
```

START_X25_INTERFACE (STAXI)

Purpose

Starts the specified X.25 Packet Level interface. The START_X25_INTERFACE command starts the X.25 packet level protocol on the X.25 trunk supported by the interface. This command also starts the underlying X.25 trunk.

Format

START_X25_INTERFACE
INTERFACE_NAME = name

Parameters

INTERFACE_NAME (IN)

The logical name of an X.25 packet level interface defined by a DEFINE_X25_INTERFACE command.

Responses

X.25 interface <name> started on trunk <trunk_name>.

--ERROR-- X.25 interface <name> already started.

--ERROR-- X.25 interface <name> is not defined for this system.

--ERROR-- X.25 interface <name> already started for trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> down. Unable to start X.25 interface <interface_name>.

--ERROR-- Trunk <trunk_name> off. Unable to start X.25 interface <interface_name>.

--FATAL-- Stream Service Error.
Error code = <error_code>.

--FATAL-- Unable to start task <entry_point_name>.

--FATAL-- X.25 interface <name> not responding -- interface unconditionally stopped.

--FATAL-- X.25 interface <name> reported error -- interface unconditionally stopped.

Remarks

For more information on X.25 Packet Level interface, refer to the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
senc c='start_x25_interface in=telenet_inf',s=ndi1

X.25 interface TELENET_INF started on trunk TELENET2.
```

STOP Commands

STOP_CIM_TEST (STOCT)

Purpose

Stops an online diagnostics test running on a Communications Interface Module (CIM) and its LIMs.

Format

STOP_CIM_TEST
DEVICE_NAME = name

Parameters

DEVICE_NAME (DN)

Physical name of the CIM, derived from its type (CIM) and its board slot number (0..7). For example, $CIM3 is the physical name for a CIM board in slot 3.

Responses

CIM test stop flag set, version <version_number>.
CIM slot number = <cim slot_number>.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- CIM test not running.

CIM slot number= <cim slot number>.

Remarks

To get the results of the CIM test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

Examples

```
senc c='stop_cim_test device_name=$cim5',s=tdi5

CIM test status
CIM slot number = 5.
PASSED on-line version <version_number> <date> <time>
pass count = <pass_count>
```

STOP_ESCI_TEST (STOET)

Purpose Stops an online diagnostics test running on an Ethernet Serial Channel Interface (ESCI).

Format **STOP_ESCI_TEST**
DEVICE_NAME = name

Parameters *DEVICE_NAME (DN)*

Physical name of an ESCI board consisting of a dollar sign ($), board type (ESCI) and its board slot number (0..7). For example, $ESCI4 is the physical name of an ESCI board in slot 4.

Responses ESCI test stop flag set, version <version_number>.
ESCI slot number = <ESCI_slot_number>.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- <Device> test not running ESCI slot number= <esci slot number>.

Remarks To get the results of the ESCI test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

Examples

```
senc c='stop_esci_test device_name=$esci6',s=north_tdi_1

ESCI test status
ESCI slot number = 6.
PASSED on-line version <version_number> <date> <time>
pass count = <pass_count>
```

STOP_LIM_TEST (STOLT)

Purpose Stops the online diagnostics running on a LIM board and its ports.

Format **STOP_LIM_TEST**
DEVICE_NAME = name

Parameters **DEVICE_NAME**

Physical name of LIM device, consisting of a dollar sign ($), board type (LIM) and slot number, as in $LIM5 (device name for LIM board in slot 5).

Responses LIM test stop flag set, version <version_number>.
CIM Slot number = <cim_slot_number>.
LIM Slot number = <lim_slot_number>.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- LIM test not running.

Remarks To get the results of the LIM test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

Examples

```
senc c='stop_lim_test device_name=$lim2',s=south_tdi

LIM test status
CIM Slot number = 5.
LIM Slot number = 2.
PASSED on-line version <version_number> <date> <time>
pass count = <pass_count>
```

STOP_LINE (STOL)

Purpose Stops communications over a communication line or a URI line.

Format **STOP_LINE**
LINE_NAME = name

Parameters **LINE_NAME (LN)**

Logical name of the line assigned by the DEFINE_LINE command that configured the line.

Responses Line <line_name> stopped.

--WARNING-- Line <line_name> already stopped.

--ERROR-- Line <name> not defined for this system.

--ERROR-- Line <name> down, hardware status indicates port is in a DOWN or OFF state.

--FATAL-- Line shutdown failure.

Examples

```
senc c='stop_line line_name=engin_bld_31',s=engin_tdi_4

Line ENGIN_BLDG_31 stopped.
```

STOP_LINE_METRICS (STOLM)

Purpose

Stops the collection and reporting of statistics at a statistics level for one or more communication lines or URI lines. Statistics are immediately reported for the stopped line statistics. Any statistics groups not specifically stopped continue to be collected and reported.

Format

STOP_LINE_METRICS
LINE_NAME = list 1..15 of name
GROUP = list 1..2 of keyword value

Parameters

NAME (N)

Logical name or names of the line or lines for which you want to stop statistics collection and reporting.

GROUP (G)

Statistics group for which you want to stop collection and reporting. Possible keyword values include SUMMARY, EXPANDED, and ALL. Default is ALL.

Responses

In the current CDCNET release, the STOP_LINE_METRICS command returns a response listing the line metrics successfully stopped, and a response listing the line metrics not stopped due to errors during command entry.

Line <line_name> <group_name> metrics stopped.
(One response for each line and group specified in the command.)

For lines that are not defined or if the metric was not started, the following lines are displayed.

Line <line_name> not defined.

Line <line_name> <group_name> metrics not started.

--FATAL-- Line <line_name> metrics shutdown failed.

Examples

```
senc c='stop_line_metrics line_name=(line_303,line_305,..
line_306,line_310),g=(summary,expanded)',s=west_tdi

Line LINE_303 summary metrics stopped.
Line LINE_303 expanded metrics stopped.
Line LINE_305 summary metrics stopped.
Line LINE_305 expanded metrics stopped.
Line LINE_306 summary metrics stopped.
Line LINE_306 expanded metrics stopped.
Line LINE_310 summary metrics stopped.
Line LINE_310 expanded metrics stopped.
```

STOP_MCI_INLINE_TEST (STOMIT)

Purpose Stops the inline diagnostics test executing on an MCI board.

Format **STOP_MCI_INLINE_TEST**
DEVICE_NAME = name

Parameters **DEVICE_NAME**

Physical name of the device to be tested, consisting of a dollar sign ($), board type (MCI, in this case), and slot number. This parameter has no default value.

Responses Stopped the MCI in line test
for device <device name>.

--ERROR-- Device <device_name> is not installed in system.

--ERROR-- Device <device_name> is not a MCI board.

--ERROR-- MCI in line test for device <device_name> is not running.

--ERROR-- MCI in line test for device <device_name>
was terminated. However, no termination response
was received from the in line diagnostics test.

Examples

```
senc c='stop_mci_inline_test dn=$mci6'

Stopped the MCI in line test
for device $mci6
```

STOP_MCI_TEST (STOMT)

Purpose Stops the online diagnostic running on the MCI.

Format

STOP_MCI_TEST
DEVICE_NAME = name

Parameters

DEVICE_NAME (DN)

Physical name of the MCI. The physical name consists of a dollar sign $, board type (MCI), and the slot number, as in $MCI6 (device name for an MCI in slot 6). There is no default value.

Responses

MCI test stop flag set, version <version number>

MCI slot number= <mci slot number>

--ERROR--

Device <device_name> not installed in system

--ERROR--

MCI test not running
MCI slot number= <mci slot number>

Examples

```
senc c=' stop_mci_test dn=$mci6'

MCI test stop flag set, version 10G2
MCI slot number= 6
```

STOP_NETWORK (STON)

Purpose Stops communications over a network solution, such as Ethernet, X.25, HDLC, or channel. For an Ethernet network, STOP_NETWORK also stops the underlying Ethernet trunk. For an X.25 network, STOP_NETWORK clears the virtual circuit underlying the network, but does not stop the Packet Level interface or X.25 trunk supporting the network. Those elements of the X.25 interface must be stopped by the STOP_X25_INTERFACE command.

Format **STOP_NETWORK**
NETWORK_NAME = name

Parameters **NETWORK_NAME (NN)**

The logical name of the network assigned by a define command. For NOS host channels, specify the trunk name for this parameter. The default trunk name is $MCI<slot>, where <slot> is the board slot number of the MCI. for trunk <trunk_name>.

--WARNING-- Network <name> already stopped for trunk <trunk_name>.

--WARNING-- The 3A Command Processor has timed-out waiting for response from SSR.
Please check network status for completion of request.

--ERROR-- Network <name> is not defined.

--FATAL-- Stream Service Error.

(See below.)
The device manager did not accept a function for the ESCI board.
HDLC SSR received error when sending command to DVM.

Examples

```
senc c='stop_network network_name=tymnet_net_1',s=ndi1

X.25 Network TYMNET_NET_1 stopped for trunk
TYMNET_TRUNK1.
```

STOP_NETWORK_METRICS (STONM)

Purpose

Stops the collection and reporting of statistics at a statistics level for one or more network solutions. Statistics are immediately reported for the stopped network statistics. Any statistics groups not specifically stopped continue to be collected and reported.

Format

STOP_NETWORK_METRICS
NETWORK_NAME = list 1..15 of name
GROUP = list 1..2 of keyword value

Parameters

NETWORK_NAME (NN)

Logical names of the network solutions for which you want to stop statistics collection and reporting.

GROUP (G)

Statistics group whose collection and reporting you want to stop. Possible keyword values include the following: SUMMARY, EXPANDED, and ALL. Default is ALL.

Remarks

In CDCNET release 1.2.5, the STOP_NETWORK_METRICS command returns a response listing the network metrics successfully stopped, and the network metrics not stopped due to errors during command entry.

Network <network_name> <group_name> metrics stopped (one response for each network and group specified in the command).

For networks that are not defined, or if the metric was not started, the following lines are displayed.

Network <network_name> not defined.

Network <network_name> <group_name> metrics not started.

--FATAL-- Network <network_name> metrics shutdown failed.

Examples

```
send_command c='stop-network_metrics ..
network_name=bld_3_ethernet,group=(summary,expanded)',s=mdi_01

Network BLD_3_ETHERNET summary metrics stopped.
Network BLD_3_ETHERNET expanded metrics stopped.
```

STOP_NP_GW (STONG) (NOS Only)

Purpose

Disconnects any application-to-application (A-to-A) connections supported by a Network Products A-to-A gateway and deletes the title or titles registered for the gateway in the CDCNET directory. STOP_NP_GW inactivates any title or titles added with the ADD_X25_OUTCALL command. These titles remain known to the gateway and are reactivated when the gateway is restarted. STOP_NP_GW both stops and cancels the Network Products A-to-A gateway. The STOP_NP_GW command essentially removes the gateway from use.

Format

STOP_NP_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name assigned to a Network Products gateway by a DEFINE_NP_GW command.

Responses

NP gateway service <gateway_name> stopped.

--ERROR-- NP gateway <name> not defined or already stopped.

--FATAL-- NP gateway <name> shutdown failed.

Examples

```
senc c='stop_np_gw gn=a_to_a_109' s=mdi2

NP gateway service A_TO_A_109 stopped.
```

STOP_NP_INTERFACE (STONI) (NOS Only)

Purpose

Stops the Network Products protocol over a mainframe channel to a NOS system and stops the underlying channel trunk protocol. The Network Products interface is addressed by its interface name.

Format

STOP_NP_INTERFACE
INTERFACE_NAME = name

Parameters

INTERFACE_NAME (IN)

The logical name of the Network Products interface assigned by a define command.

Responses

NP_interface <interface_name> stopped.

--WARNING-- NP interface <interface_name> already stopped.

--ERROR-- NP interface <interface_name> is not defined.

--FATAL-- NP interface <interface_name> command processor has timed-out waiting for a response from the NP interface task.

--FATAL-- Unable to stop the NP interface <interface_name>. Unable to send ITM to NP interface task.

Examples

```
senc c='stop_np_interface in=cyber_109',s=mdi2

NP interface CYBER_109 stopped.
```

STOP_NP_TERMINAL_GW (STONTG)

Purpose

Disconnects any terminal-to-application connections supported by a Network Products (NP) interactive gateway, and deletes the titles registered for the gateway in the CDCNET directory. The command removes the NP terminal gateway from use.

Format

STOP_NP_TERMINAL_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of an NP terminal gateway, assigned by a DEFINE_NP_TERMINAL_GW command that configured the gateway.

Responses

NP terminal gateway <name> stopped.

--ERROR-- NP terminal gateway name not defined or already stopped.

--FATAL-- NP terminal gateway <name> shutdown failed.

Examples

```
senc c='stop_np_terminal_gw gw=ivt109'

NP terminal gateway IVT109 stopped.
```

STOP_PASSTHROUGH_SERVICE (STOPS)

Purpose Stops the Interactive Passthrough Service.

NOTE

This command terminates all existing passthrough connections in the DI.

Format **STOP_PASSTHROUGH_SERVICE**

Parameters None.

Responses Passthrough Service stopped.

--ERROR-- Passthrough Service not defined or already stopped.

Examples

```
senc c='stop_passthrough_service'

Passthrough Service stopped.
```

STOP_PORT_TEST (STOPT)

Purpose Stops an online diagnostics test running on an individual LIM port.

Format **STOP_PORT_TEST**
DEVICE_NAME = name

Parameters **DEVICE_NAME (DN)**

Physical name of the port, consisting of a dollar sign ($) board type (LIM) its slot number, the keyword PORT and port number. For example, $LIM3_PORT1 is the device name for port 1 on the LIM board in slot 3.

Responses PORT test stop flag set, version <version_number>.
CIM Slot number = <slot_number>.
LIM Slot number = <slot_number>.
PORT Slot number = <slot_number>.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Port test status

CIM slot number= <cim slot number>
LIM slot number= <lim slot number>
Port number= <port number>

Remarks To get the results of the LIM test, send the DISPLAY_TEST_STATUS command to the DI that contains the device being tested.

Examples

```
senc c='stop_port_test device_name=$lim3_port2',s=tdi_5

PORT test status
CIM slot number = 5
LIM slot number = 3
Port slot number = 2
PASSED on-line version <version_number> <date> <time>
pass count = <pass_count>.
```

STOP_PROCESS_METRICS (STOPM)

Purpose

Stops the collection and reporting of statistics at a statistics level for a software process. Statistics are immediately reported for the stopped process statistics. Any statistics groups not specifically stopped continue to be collected and reported.

Format

STOP_PROCESS_METRICS
PROCESS = list 1..15 of name
GROUP = list 1..3 of keyword value

Parameters

PROCESS (P)

Logical names of the software processes for which you want to stop statistics collection and reporting. The following software process names are supported:

Intranet
XNS_internet
Generic_transport
XNS_transport
Session
NP_IVT_GW
Routing
File_access
Directory
Log_Support_Application (source log)
Independent_Log_ME (record log)
Command
OSA (Operator Support Application)
LCM (Line Control Module)
System
DOD_Internet
TCP
Telenet_Inteface

GROUP (G)

Statistics group for which you want to stop collection and reporting. Possible keyword values include the following: SUMMARY, EXPANDED, DEBUG, and ALL. Default is ALL.

Responses

In the current CDCNET release, the STOP_PROCESS_METRICS command returns a response listing the process metrics successfully stopped and the process metrics not stopped because of errors during command entry.

<process_name> <group_name> metrics stopped.
(One response for each process and group specified in the command).

For processes that are unknown or if the metric was not started the following lines are displayed:

The following line displays when the process is unknown to statistics.

Process <process_name> unknown to statisics

The following line displays when the group has never started.

<process_name> <group_name> metrics not started

The following line displays when an internal error occurs.

<process_name> metrics shutdown failed.

Examples

In this example, summary and expanded metrics for the XNS Transport process are stopped. Debug statistics are not stopped, and they will continue to be collected and reported after the summary and expanded metrics are stopped.

```
send_command c='stop_process_metrics p=xns_transport ..
g=(summary,expanded)',s=mdi_3

XNS_TRANSPORT summary metrics stopped.
XNS_TRANSPORT expanded metrics stopped.
```

STOP_SERVER_TELNET_GW (STOSTG)

Purpose Stops the host terminal gateway service. The gateway terminates any established connections and stops listening for new connections. This command reverses the effect of a START_SERVER_TELNET_GW command.

Format **STOP_SERVER_TELNET_GW**
GATEWAY_NAME = name

Parameters **GATEWAY_NAME (GN)**

The logical name of the server TELNET host gateway defined by a DEFINE_SERVER_TELNET_GW command.

Responses Server TELNET gateway <gateway_name> is stopped.

--WARNING-- Server TELNET gateway <gateway_name> is already stopped.

--ERROR-- Server TELNET gateway <gateway_name> is not defined.

Examples

```
senc c='stop_server_telnet_gw gateway_name=gw_to_cyber',s=ndi1

Server TELNET gateway GW_TO_CYBER is stopped.
```

STOP_TCPIP_GW (STOTG)

Purpose

Stops the TCP/IP application interface gateway. The gateway terminates any established connections and deregisters (clears) all titles associated with this gateway interface. This command reverses the effect of a START_TCPIP_GW command.

Format

STOP_TCPIP_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the TCP/IP application gateway defined by a DEFINE_TCPIP_GW command.

Responses

TCP/IP gateway <gateway_name> is stopped.

--ERROR-- TCP/IP gateway <gateway_name> is not defined.

--FATAL-- TCP/IP gateway was unable to deregister title <title>.

--WARNING-- TCP/IP gateway <gateway_name> is already stopped.

Examples

```
senc c='stop_tcpip_gw gateway_name=ftp_gw',s=ndi1

TCP/IP gateway FTP_GW is stopped.
```

STOP_URI_TEST (STOUT)

Purpose Stops the online diagnostic test running on a URI.

Format **STOP_URI_TEST**
DEVICE_NAME = name

Parameters **DEVICE_NAME (DN)**

Physical name of the URI, consisting of a dollar sign ($), board type (URI), and its slot number.

Responses URI test stop flag set, version <version number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>

--ERROR-- Device <device_name> not installed in system.

--ERROR-- URI test not running
CIM slot number= <cim slot number>
URI slot number= <uri slot number>

Examples

```
senc c='stop_uri_test dn=$uri3'

URI test stop flag set, version 2301
CIM slot number= 3
URI slot number= 5
```

STOP_USER_TELNET_GW (STOUTG)

Purpose

Stops the user TELNET interactive terminal gateway service. The gateway terminates any established connections and deregisters (clears) titles associated with this gateway interface. No new connections can be established. This command reverses the effect of a START_USER_TELNET_GW command.

Format

STOP_USER_TELNET_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the user TELNET gateway defined by a DEFINE_USER_TELNET_GW command.

Responses

User TELNET gateway <gateway_name> is stopped.

--WARNING-- User TELNET gateway <gateway_name> is already stopped.

--ERROR-- User TELNET gateway <gateway_name> is not defined.

Examples

```
senc c='stop_user_telnet_gw gateway_name=vax_gw',s=ndi1

User TELNET gateway VAX_GW is stopped.
```

STOP_X25_ASYNCTIP (STOXA)

Purpose

Stops X.25 asynchronous TIP service for the specified X.25 trunks. Disconnects any active terminal connections through the X.25 asynchronous TIP for the specified trunk.

Format

STOP_X25_ASYNCTIP
TRUNK_NAME = list of 1..32 of name

Parameters

TRUNK_NAME (TN)

Logical name of one or more X.25 trunks for which X.25 asynctip service is to be stopped.

Responses

X.25 AsyncTip support stopped for specified trunks.

X.25 AsyncTip support already stopped for trunk <trunk_name>.

--ERROR-- X.25 AsyncTip support not defined for trunk <trunk_name>.

--ERROR--Duplicate trunk name <trunk_name> specified.

Examples

```
senc c='stop_x25_asynctip trunk_name = telenet_2'

X.25 AsyncTip support stopped for specified trunks.
```

STOP_X25_GW (STOXG)

Purpose Disconnects any application-to-application connections supported by the X.25 transparent gateway and deletes the title(s) registered for the gateway in the CDCNET directory. The STOP_X25_GW command removes the X.25 gateway from use.

Format

STOP_X25_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name assigned to an X.25 gateway by a DEFINE_X25_GW command.

Responses

X.25 gateway <name> stopped.

--WARNING-- X.25 gateway <name> is already stopped.

--ERROR-- X.25 gateway <name> is not defined.

Examples

```
senc c='stop_x25_gw gateway_name=telenet_gw',s=ndi1

X.25 gateway TELENET_GW stopped.
```

STOP_X25_INTERFACE (STOXI)

Purpose

Stops the specified X.25 Packet Level interface. The STOP_X25_INTERFACE command stops the X.25 Packet Level protocol on the X.25 trunk supported by the interface. The STOP_X25_INTERFACE command also stops the underlying X.25 trunk.

Format

STOP_X25_INTERFACE
INTERFACE_NAME = name

Parameters

INTERFACE_NAME (IN)

The logical name assigned to an X.25 interface by a DEFINE_X25_INTERFACE command.

Responses

X.25 interface <name> stopped on trunk <trunk_name>.

--WARNING-- X.25 interface <name> already stopped.

--ERROR-- X.25 interface <name> is not defined for this system.

Examples

```
senc c='stop_x25_interface interface_name=telenet_if',s=ndi1

X.25 interface TELENET_IF stopped on trunk TELENET2.
```

Miscellaneous Commands

DELETE_X25_GW_OUTCALL (DELXGO)

Purpose

Deletes an X.25 gateway outcall title from the specified gateway. The gateway must have been previously defined.

Format

DELETE_X25_GW_OUTCALL
TITLE = string 1..255
GATEWAY_NAME = <name>

Parameters

TITLE (T)

Specifies the title that your CDNA applications can use to access a particular remote application through the gateway. The title supports calls from CDNA systems to remote systems accessed through the X.25 network.

GATEWAY_NAME (GN)

Specifies the name of the X.25 gateway that provides access to the remote application.

Responses

X.25 gateway title <title> deleted.

--ERROR-- X.25 gateway title <title> was not found.

--ERROR--X.25 gateway <name> is not defined.

Examples

```
senc c='=delete_x25_gw_outcall t=''PTFS$FOREIGN'''

X.25 gateway title PTFS$FOREIGN deleted.
```

HELP

Purpose Performs the same function as the DISPLAY_COMMAND_LIST command. Refer to the DISPLAY_COMMAND_LIST command previously described in this chapter.

Format **HELP**

Responses <Alphabetical list of all network commands. See example.>

Examples

```
senc c='help'

add_np_gw_outcall                                     add_x25_gw_outcall
      .                                                      .
      .                                                      .
      .                                                      .
unload_module                                         write_terminal_message
```

KILL_SYSTEM (KILS)

Purpose

Shuts off a DI's system hardware clock without a graceful shutdown. You must reload the DI software. You may optionally request a dump of DI memory contents.

NOTE

Notify all active users that they will be disconnected from CDCNET services by sending a message using the WRITE_TERMINAL_MESSAGE command.

Format

KILL_SYSTEM
DUMP = boolean

Parameters

DUMP (D)

Requests a full DI memory dump before reload. Possible parameter values include YES and NO. Default is NO.

Remarks

The KILL_SYSTEM command is one of the error conditions defined for DIs. KILL_SYSTEM with a dump is assigned DI error condition code 32 hexadecimal; KILL_SYSTEM without a dump is assigned error condition code 33 hexadecimal. These error conditions are significant in the configuration process for a DI, as they can be used when defining the loading and dumping conditions for a DI. For more information, refer to the following sections of the CDCNET Configuration and Site Administration Guide, appendix F (DI Reset Codes) and the descriptions of DEFINE_BOOT_DEFAULTS and DEFINE_EXCEPTION_SYSTEM.

Responses

System being reset and reloaded.

Examples

```
senc c='kill_system',s=north_tdi_1

System being reset and reloaded.
```

LOAD_MODULE (LOAM)

Purpose

Loads a specified software module and optionally sets the module load status to retained. If the software module is already loaded, the LOAD_MODULE command only sets the retain status for the module; it does not guarantee that a new copy of the module will be loaded. A retained module will not be unloaded to recover system memory resources even if the module is unused and memory resources are scarce.

Format

LOAD_MODULE
MODULE = name
RETAIN = boolean

Parameters

MODULE (M)

The name of the desired software module.

RETAIN (R)

The retain status for the module. Default is YES, retain.

Responses

Module <module> loaded.

Module <module> loaded and retained.

--WARNING-- Module <module> previously retained.

--WARNING-- Declaration mismatch from module <module>.

--ERROR-- Module <module> was not found in directory.

--ERROR-- Module for entry point <entry point> was not found.

--FATAL-- On-line loader not included in boot file.

--FATAL-- Unable to access file load service.

--FATAL-- Not enough memory is currently available to load module <module>.

--FATAL-- File access error is unrecoverable.

--FATAL-- Duplicate definition of entry point <entry point> encountered.

--FATAL-- Identification record expected for module <module>.

--FATAL-- Unrecognizable record in module <module>.

--FATAL-- Premature EOF encountered on module <module>.

--FATAL-- Object text version must be <version>, but is <version>.

--FATAL-- Object text record too long in module <module>.

Remarks

To display the modules currently loaded in a DI, send the DISPLAY_SOFTWARE_LOAD_STATUS (see command description in this chapter) command to the DI. An alternative method is to use the NPA report LOADRP1 to identify modules loaded per DI. Refer to the NPA manual for more information on generating LOADRP1.

Examples

This example shows a software module containing the DISPLAY_HARDWARE_STATUS (DISHS) command processor being loaded into the DI by the LOAD_MODULE command. This is done so that the display hardware status command processor is loaded in the DI and retained there, so that when the DISHS command is entered, it may be processed more quickly than it would be if the processor had to be accessed and loaded using the Online Loader.

```
send_command s=mdi_1,c='load_module module=display_hardware_status'

Module DISPLAY_HARDWARE_STATUS loaded and retained.
```

This example shows the command processor from example 1 being loaded with the RETAIN parameter set to NO. The command processor is loaded into the DI, but if it is not used and the memory it occupies is needed, it does not remain.

```
senc c='load_module m=display_hardware_status r=no',s=mdi_1

Module DISPLAY_HARDWARE_STATUS loaded.
```

SET_DATE_AND_TIME (SETDAT)

Purpose

Sets the master date and time for a catenet. For NOS-based CDCNET environments, the master date and time is maintained by one DI in the network that is configured as the clocking_system DI. A clocking_system DI contains the Independent Clock Management Entity. For NOS/VE-based CDCNET environments, the master date and time is maintained in a NOS/VE host. For NOS environments, this command must be sent to the clocking_system DI.

Each CDCNET DI reports date and time in command responses, logs and alarms. Each DI also contains a Dependent Clock ME, which obtains the master Catenet clock from the clocking_system DI (or from the master clock on the NOS/VE host in NOS/VE environments). When the correct date and time is set, you can send the SYNCHRONIZE_CLOCK command to each DI in the network (see SYNCHRONIZE_CLOCK command description), to reset each DI's clock to the master date and time.

Format

SET_DATE_AND_TIME
DATE = string
TIME = string
DATE_FORMAT = keyword value
TIME_FORMAT = keyword value

Parameters

DATE (D)

Current date, represented in the format specified by the DATE_FORMAT parameter (see parameter description). If this parameter is not entered, the CDCNET date is not changed. The allowable range for the day component is dependent on the month and year. Range for January, March, May, July, August, October, December is 1..31; for April, June, September, November, 1..30; and for February, 1..28 or 1..29. The allowable range for the month component is 01..12. If the DATE_FORMAT selected is ISO, the ISO year range is 1900..2155.

TIME (T)

Current time, represented in the format specified by the TIME_FORMAT parameter (see parameter description). If time is not entered, the current time is used. The allowable range for the minute and second components is 00..59. If the TIME_FORMAT selected is AMPM, the hour component may be in the range 01..12, otherwise the range is 00..23.

DATE_FORMAT (DF)

Specifies how date information will be specified. Allowed keyword values include the following, using as an example a date of November 1, 1985, and dd for day, mm for month, and yy for year.

Keyword Value	Format	Example
MDY	mm/dd/yy	11/01/85
DMY	dd/mm/yy	01/11/85
ISO	yyyy-mm-dd	1985-11-01

Default is DMY.

TIME_FORMAT (TF)

Specifies how time information will be specified. Allowed keyword values include the following, using as an example a time of 2:41 PM, and hh for hour, mm for minute, ss for second, and XX for AM or PM identifier.

Keyword Value	Format	Example
AMPM	hh:mm XX	2:41 PM
HMS	hh:mm:ss	14:41:38

Default is HMS.

Responses

Master clock for catenet set.

(Followed by date and time in selected format. See example.)

--WARNING-- Master clock for catenet set

(Followed by date and time in selected format)
Power on reset <text> used, please correct.

--WARNING-- Master clock for catenet set

(Followed by date and time in selected format)
Power on reset date and time used, please correct.

--ERROR-- Alphabetic character in date: <text>.

--ERROR-- Alphabetic character in time: <text>.

--ERROR-- Day value <text> out of range.

--ERROR-- Day value <text> out of range for month <text>, year <text>.

--ERROR-- Month value <text> out of range.

--ERROR-- Year value <text> out of range.

--ERROR-- Hour value <text> out of range.

--ERROR-- Minute value <text> out of range.

--ERROR-- Second value <text> out of range.

--ERROR-- Expecting date in format <text>, found <text>.

--ERROR-- Expecting time in format <text>, found <text>.

--ERROR-- Independent clock ME not installed in system.

Remarks

The clocking_system DI is configured by the CLOCKING_SYSTEM parameter on the DEFINE_SYSTEM command. To determine which DI is configured to be the clocking_system, send the DISPLAY_SYSTEM_OPTIONS (DISSO) command to each DI. Specify the display option CLOCKING_SYSTEM, as shown in the following example.

```
SEND_COMMAND SYSTEM=di_name,COMMAND='DISPLAY_SYSTEM_OPTIONS..
DISPLAY_OPTION=CLOCKING_SYSTEM'
```

The DI that contains the master clock returns the following response.

```
clocking_system = yes
```

If any component of the date or time is omitted, the corresponding component of the current date or time is used. For example, if you enter df=dmy,d="//86", the year will change to 1986, but the current day and month will not be changed. Leading zeros may be omitted from any component number, provided that the component is preceded by a delimiter or a letter. The following are valid delimiters.

blank	space
/	slant
-	hyphen
:	colon

Examples

```
senc c='set_date_and_time d=''24/11/85'',..
t=''08:25:49''',s=main_mdi

Master clock for catenet set
24/11/85 08:25:49
```

SYNCHRONIZE_CLOCK (SYNC)

Purpose

Sets a DI's date and time to the master date and time for a catenet.

The master date and time is maintained by a DI or NOS/VE system that contains the network-wide clock management function. The master date and time for the catenet is set in a DI by the SET_DATE_AND_TIME command and on the NOS/VE system according to the system's date and time. When the SYNCHRONIZE_CLOCK command is sent to a DI, the DI's clock is set to the master date and time.

Format

SYNCHRONIZE_CLOCK

Remarks

System clock synchronized.

--FATAL-- Unable to access master clock through Independent Clock M-E.

--FATAL-- Unable to synchronize system clock, version number mismatch.

--FATAL-- Unable to synchronize system clock, retry limit reached.

Examples

```
send_command c='synchronize_clock',s=engin_tdi_1

System clock synchronized.
```

UNLOAD_MODULE (UNLM)

Purpose

Marks a module as a candidate for unloading from a DI. This command clears the retain flag from the module so that when the module is no longer used, the module can be unloaded if memory is needed.

An unloaded module may be reused by the system if the module remains resident in a DI. UNLOAD_MODULE does not guarantee that the module is immediately unloaded or that a new copy of the unloaded module is used.

Format

UNLOAD_MODULE
MODULE = name

Parameters

MODULE (M)

The name of the desired software module.

Responses

Module <module> retain removed.

--ERROR-- Module <module> not currently loaded.

--ERROR-- Module <module> not previously retained.

Remarks

To display the modules currently loaded and/or marked for unloading in a DI, send the DISPLAY_SOFTWARE_LOAD_STATUS (see command description in this chapter) command to the DI.

Examples

```
senc c='unload_module m=display_hardware_status',s=engin_tdi

Module CMD_DISPLAY_HARDWARE_STATUS retain removed.
```

WRITE_TERMINAL_MESSAGE (WRITM)

Purpose

Sends a message to an interactive terminal or group of terminals, including the control consoles for batch workstations. This command allows you to send informative or warning messages to network users or to respond to a network user's request.

You may choose the terminals to which the message is sent by three attributes: line name, terminal device name, or connected service. Specifying these attributes limits the number of terminals receiving a message to those terminals that match the specified attributes.

If you do not specify any attributes with the command and message, then all terminals with at least one active session receive the message.

You can restrict the number of terminals receiving a message by sending the WRITM command only to the DIs to which the desired terminals are attached.

Format

WRITE_TERMINAL_MESSAGE
MESSAGE = list 1..15 of string
LINE_NAME = list 1..15 of name
DEVICE_NAME = list 1..15 of name
SERVICE_NAME = list 1..15 of name

Parameters

MESSAGE (M)

Text of the message to the terminal user. This message must be enclosed by apostrophes. Since this command will be sent as a string value within SEND_COMMAND, you must begin and end the message with two consecutive apostrophes so that the message will be distinguished as a string value within another string value. For a list of strings, each string is output as one display line. The message may be any text up to 245 characters long. For example, the text ('Please log off by 14:00','Network temporarily down for diagnostics') produces the following output:

```
Please log off by 14:00
Network temporarily down for diagnostics
```

LINE_NAME (LN)

Logical name(s) of the line or lines to receive the message.

DEVICE_NAME (DN)

Logical name(s) of the terminal or terminals to receive the message.

SERVICE_NAME (SN)

Name of the service or services to which terminals must be connected if they are to receive the message.

Responses Message written.

--WARNING-- No terminal matched attributes, message not written.

--FATAL-- Message output process failed.

NOTE

A success response is returned even if no terminals are active, if no terminal interface program (TIP) is installed in the DI to which the terminal is connected, or if the terminal user has disabled output of operator messages.

Remarks At an interactive terminal, the message begins on the next line following the current cursor position. If there is output ready for a terminal from a working connection, the message will be inserted in the output. If the terminal has multiple working connections, the message will appear immediately, regardless of the connection currently in use. If the user disables output of operator messages, the messages sent to the terminal are discarded, and are not retained for display at a later time.

Each message begins with the date and time from the DI to which the terminal is connected. A message appears in the following format, where the message text may be one or more lines of text.

```
yy/mm/dd hh.mm.ss   FROM NETWORK OPERATOR
<message te>
```

Messages are sent from terminal users to the network operator by the REQUEST_NETWORK_OPERATOR (REQNO) terminal user command.

Examples

```
send_command c='write_terminal_message,..
m=(''New communications configuration tomorrow'',''Network down ..
until 10:00.'')',s=tdil

Message written.
```

Configuration Commands

DI Configuration Procedure Commands - Common to NOS/VE and NOS

This section contains descriptions of commands used in DI system configuration procedures in both NOS/VE and NOS environments. For commands used only in a NOS or NOS/VE environment, refer to the sections in this chapter entitled, DI Configuration Procedure Commands–NOS Only, and DI Configuration Procedure Commands–NOS/VE Only.

ADD_X25_GW_OUTCALL (ADDXGO)

Purpose

Defines a gateway outcall definition. Outcall is from the perspective of the CDCNET network; that is, the call is going out of the CDCNET network. The outcall information is used to generate the proper call request into the foreign network. An X.25 gateway outcall consists of a CDNA title, outcall addressing, and connection parameters associated with an X.25 gateway. NOS/VE applications or other gateways translate on this type of title to make direct outgoing calls on X.25 without the application specifying X.25 addressing. Refer also to the DEFINE_X25_GW command description in this chapter.

Format

ADD_X25_GW_OUTCALL
TITLE = name
REMOTE_DTE_ADDRESS = string 1..15
PROTOCOL_ID = 2..255
GATEWAY_NAME = name
LOCAL_DTE_ADDRESS = string 1..15
FACILITIES = string 1..63
USER_DATA = string 2..248

Parameters

TITLE (T)

The title that CDNA applications can use to access a particular remote application through this gateway. The title is used to support calls from CDNA systems to remote systems accessed through the X.25 network.

REMOTE_DTE_ADDRESS (RDA)

The X.25 address of the destination X.25 system. This parameter is specified as a string of digits 0 through 9.

PROTOCOL_ID (PI)

The protocol identifier as required by the destination X.25 system. Octets 2 through 4 are set to zero by the gateway.

GATEWAY_NAME (GN)

The name of the X.25 gateway which provides access to the remote application. The gateway must be previously defined. If this command is specified in a configuration file, the default value for this parameter is the previously-defined X.25 gateway name. If this command is entered by the network operator through the Network Operator Utility (NETOU), this parameter is required.

LOCAL_DTE_ADDRESS (RDA)

The X.25 address of a local X.25 trunk. The call request is attempted over the X.25 trunk with the matching dte_address. The call is rejected if no matching trunk is found. This parameter is specified as a string of digits 0 through 9. If this parameter is not specified, the X.25 Packet Level will select a trunk to make the call request.

FACILITIES (F)

The facilities options as defined by the X.25 CCITT protocol. For information on X.25 facilities options, refer to CCITT Recommendation X.25. This parameter is specified as an even-numbered string of hexadecimal digits.

USER_DATA (UD)

An even-numbered string of hexadecimal digits. This parameter value is added to the beginning of any "real" user data from the session indication and the concatenated string is then placed into the USER_DATA field of the X.25 call. The call is rejected if the concatenated string exceeds the field size. The maximum field size is 124 octets with the fast select facility, and 12 octets without the fast select facility.

Responses

X.25 gateway title <title> added.

--ERROR-- An X.25 gateway is not defined.

--ERROR-- Remote_dte_address can not include <string>. A remote_dte_address may include only digits 0 through 9.

--ERROR-- Local_dte_address can not include <string>. A local_dte_address may include only digits 0 through 9.

--ERROR-- Facilities can not include <string>. Facilities may include only hexadecimal digits 0 thru 9 and a thru f.

--ERROR-- Facilities can only have an even number of hexadecimal digits.

--ERROR-- User_data can not include <string>. User data may include only hexadecimal digits 0 thru 9 and a thru f.

--ERROR-- User_data can only have an even number of hexadecimal digits.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
add_x25_gw_outcall title=PTFS$FRN,..
remote_dte_address='3401',protocol_id=0c2(16)

X.25 gateway title PTFS$FRN added.
```

CHANGE_SERVICE_DISPLAY (CHASD)

Purpose Manages the list of services that are displayable in the service availability display. The effects of multiple change commands is cumulative. The initial list of services is empty, that is, if no CHANGE_SERVICE_DISPLAY command is entered, no services are displayed in the DISPLAY_SERVICES terminal user command.

Format **CHANGE_SERVICE_DISPLAY**
ADD_SERVICES = list 1..16 of name
DELETE_SERVICES = list 1..16 of name or keyword value
STATUS_INTERVAL = 1..60 or keyword value

Parameters *ADD_SERVICES or ADD_SERVICE (AS)*

The list of interactive service names which are to be added to the list of services displayable by the DISPLAY_SERVICES terminal user command.

DELETE_SERVICES or DELETE_SERVICE (DS)

The list of interactive service names which are to be deleted from the list of services displayable by the DISPLAY_SERVICES terminal user command. If the service is included in both the ADD_SERVICES and DELETE_SERVICES parameters, the DELETE_SERVICES parameter takes precedence. The keyword ALL specifies the deletion of all services from the list.

STATUS_INTERVAL (SI)

The status of each displayable service is updated when the first DISPLAY_SERVICES command is entered. The CREATE_CONNECTION command also updates the status of a displayable service. If status_interval has not expired, the status of a displayable service is not updated when the next DISPLAY_SERVICES command is entered. The interval is in units of minutes. Default is 5 minutes.

The keyword value INFINITE specifies that the interval never expires, and that the status of a service is updated only as a result of a CREATE_CONNECTION command.

The CREATE_CONNECTION command updates the status only for the service name specified on the CREATE_CONNECTION command. That is, the status of alternate names for the same interactive service are not automatically updated. As a result, conflicts in the status of an interactive service known by multiple service names may show up when the service status is displayed.

Responses Services added to the displayable list.

<service_name>
:
<service_name>

Services deleted from the displayable list.

<service_name>
:
<service_name>

--ERROR-- Service <service_name> not in displayable list.

--ERROR-- Service <service_name> already in displayable list.

--ERROR-- No services defined in displayable list.

--FATAL-- Insufficient resources to change displayable list.

Examples

```
change_service_display add_service=veiaf
```

CHANGE_SERVICE_DISPLAY_TEXT (CHASDT)

Purpose Defines text to be displayed in the service availability display. This text is displayed when a terminal user enters the DISPLAY_SERVICES command.

Format

CHANGE_SERVICE_DISPLAY_TEXT
SERVICE = list 1..16 of name or keyword value
TEXT = list 1..4 of string 1..72
DOWN_TEXT = list 1..4 of string 1..72
TEMPORARY_DOWN_TEXT = list 1..4 of string 1..72

The TEXT, DOWN_TEXT, and TEMPORARY_DOWN_TEXT parameter definitions imply that each parameter value can be four 72-character strings. Since CDCNET commands are restricted to a total of 256 characters, it is not possible to use the full range of these parameters. The TEXT, DOWN_TEXT, and TEMPORARY_DOWN_TEXT parameters for the same service name can be specified on separate CHASDT commands.

Parameters

SERVICE or SERVICES (S)

The list of interactive service names for which the text applies. The keyword value ALL specifies that the text applies to all interactive services. If multiple services are specified, the same text applies to each service.

TEXT (T)

The text to be displayed if a service is up or busy. This text appears if the service is down and if no DOWN_TEXT or TEMPORARY_DOWN_TEXT is defined. There can be up to 4 lines of text.

DOWN_TEXT (DT)

The text to be displayed when a service is down. There can be up to 4 lines of DOWN_TEXT. It appears only if no TEMPORARY_DOWN_TEXT is defined.

TEMPORARY_DOWN_TEXT (TDT)

The text to be displayed when a service is down. There can be up to 4 lines of TEMPORARY_DOWN_TEXT. It is deleted when the service changes from down to up status. One use of this parameter is to send the CHASDT command through NETOU to enter messages for down services.

Responses Services information changed for services.

<service_name>
⋮
<service_name>

--ERROR-- Service <service_name> not in displayable list.

--ERROR-- No services defined in displayable list.

--FATAL-- Insufficient resources to change displayable list.

Examples

```
change_service_display_text service=veiaf ..
text='Call ext. 9111 if you are having problems.'
```

DEFINE_CHANNEL_TRUNK (DEFCT)

Purpose

Defines the channel level interface to a NOS or NOS/VE host. NOS/VE host channel trunks can also be configured as network solutions. Refer to the DEFINE_CHANNEL_NET command description in this chapter.

Format

DEFINE_CHANNEL_TRUNK
SLOT = 0..7
TRUNK_NAME = name
UPLINE_MESSAGE_TIMEOUT = 2..64

Parameters

SLOT (S)

The number of the physical board slot which houses the MCI board. If only one MCI board exists in the DI, then this parameter is optional.

TRUNK_NAME (TN)

The logical name of the channel trunk. The default name is constructed using the SLOT parameter, as in $MCI2.

UPLINE_MESSAGE_TIMEOUT (UMT)

The timeout value, specified in seconds, that the MCI software waits for an up-line queued message to be picked up by the host. If this value is exceeded, the interface is considered down and recovery is attempted. Default is 20.

Responses

CHANNEL trunk <trunk_name> defined.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- Board slot <slot_number> does not contain a CHANNEL board.

--ERROR-- The Device Interface does not contain a CHANNEL board.

--ERROR-- The Device Interface contains more than one CHANNEL board -- the slot must be specified.

--FATAL-- Not enough memory is currently available for required table space.

--ERROR-- Device already owned.
Card slot = <slot_number>.

--ERROR-- Device state not on.
Card slot = <slot_number>.

Examples

```
define_channel_trunk slot=2,trunk_name=channel_trunk_2
```

DEFINE_DEVICE_OUTCALL_SERVICE (DEFDOS)

Purpose

Installs the Device Outcall Service in a DI. This command should be present in the configuration files of all DIs that have devices which are to be configured as candidates to receive connections from host applications.

Format

DEFINE_DEVICE_OUTCALL_SERVICE
TITLE = name

Parameters

TITLE (T)

Specifies the title of the device outcall service. Devices connect to the device outcall service using a CREC command, with a SERVICE_NAME parameter equal to the value of this parameter. The default value is DEVICE_OUTCALL.

Responses

--INFORMATIVE-- Device Outcall Service <title> defined and started.

--ERROR-- Device Outcall Service previously defined.

--FATAL-- Not enough memory is currently available for required table space.

Remarks

The NOS/VE application Desktop/VE uses the CDCNET Device Outcall Service. The DEFINE_DEVICE_OUTCALL_SERVICE command is only available for use with Desktop/VE. The default value of the TITLE parameter (DEVICE_OUTCALL) must be used.

Examples

```
define_device_outcall_service
```

DEFINE_ETHER_NET (DEFEN)

Purpose

Configures a CDCNET Ethernet network solution using a previously defined Ethernet trunk. This command is required only in DIs that connect to an Ethernet network solution. If a TDI contains only one Ethernet trunk, you may omit this command from its system configuration procedure, since the TDI determines the information required to define the network solution by the DI load process.

Format

DEFINE_ETHER_NET
TRUNK_NAME = name
NETWORK_ID = 1..7FFFFFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
MULTICAST_NETWORK = boolean
ROUTING_INFO_NETWORK = boolean
CONGESTED_THRESHOLD = 20..255
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..500000

Parameters

TRUNK_NAME (TN)

Logical name of the Ethernet trunk to be used for the network solution. The Ethernet trunk with this name must be configured by the DEFINE_ETHER_TRUNK command before this command executes.

NETWORK_ID (NI)

CDCNET network identification number of the Ethernet network solution. This number must be unique within the catenet.

NETWORK_NAME (NN)

Logical name of the network solution that is to be used in subsequent commands referring to the network solution. The default name is constructed from the NETWORK_ID parameter, using the format $NET_network_id. network_id is the network identification number expressed in decimal, as in $NET_200.

COST (C)

Relative cost of the network solution as a path for routing data through the network. Default is OA hexadecimal.

RELAY_ALLOWED (RA)

Indicates whether relay is allowed through this network solution. Possible values are TRUE, relay allowed; and FALSE, relay not allowed. Default is TRUE.

MULTICAST_NETWORK (MN)

Indicates whether or not the network solution is a multicast network. This parameter does not have to be specified in CDCNET release 1.2.5. Possible values are TRUE and FALSE. Default is TRUE.

ROUTING_INFO_NETWORK (RIN)

Indicates whether or not the network solution carries CDCNET routing information. This parameter does not have to be specified in CDCNET release 1.2.5. Its use is not recommended. Possible values are TRUE and FALSE. Default is TRUE.

CONGESTED_THRESHOLD (CT)

For this release and future releases, this parameter is ignored.

START (S)

Specifies whether or not the network solution should start when configuration completes. Possible values are TRUE, start; and FALSE, do not start. Default is TRUE.

ARCHITECTURE_TYPE (AT)

Specifies the network architecture that this network supports. Allowed architecture types are CDNA and DOD. The DOD parameter value is currently not supported.

OUTPUT_QUEUE_LIMIT (OQL)

Specifies, in bytes, the maximum amount of data which is retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. The newest output messages are discarded first if messages need to be discarded.

The default value depends on the cost of the network (see COST parameter). If the cost is 6FA(16) or greater, then the default output queue limit is 30000 bytes. Otherwise, the default value is 60000 bytes.

Responses

Ethernet network <network_name> defined for trunk <trunk_name>.

Ethernet network <network_name> defined and started for trunk <trunk_name>.

--WARNING-- The value specified for the network_id, <value>, is greater than 65535 (0ffff(16)). Future CDCNET releases will not support a Network_id greater than 65535 (0ffff(16)).

--WARNING-- The 3A Command Processor has timed out waiting for a response from the SSR.
Please check network status for completion of request.

--ERROR-- Network <network_name> already defined for trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not an ETHERNET trunk.

--ERROR-- Network name <network_name> already defined.

--ERROR-- Network id <network_id> already defined.

--ERROR-- Trunk <trunk_name> down.

Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> off.

Unable to start network <network_name>.

--FATAL-- Not enough memory currently exists for required table space.

--FATAL-- Unable to start task <entry_point_name>.

--FATAL-- Stream service error.

The device manager did not accept a function for the ESCI board.

--FATAL-- Stream service error.

Unable to initialize ESCI board.

Examples

```
define_ether_net trunk_name=ether1,network_id=001(16),..
  network_name=net1
```

DEFINE_ETHER_TRUNK (DEFET)

Purpose

Prepares an Ethernet cable to serve as a CDCNET trunk. This command is required only in DIs that are to be configured with an Ethernet trunk. Ethernet trunks that are used to load DIs are predefined by the DI software. If you enter this command when a trunk is already defined, you receive an error message informing you of this condition. If a TDI contains only one Ethernet trunk, you may omit this command from its system configuration procedure, since the TDI determines the information required to define the trunk by the DI load process.

Format

DEFINE_ETHER_TRUNK
SLOT = 0..7
TRUNK_NAME = name
MAX_FRAME_SIZE = 1514
INTERFRAME_SPACING = 0..255

Parameters

SLOT (S)

Number of the slot that houses the Ethernet Serial Channel Interface (ESCI) board in the DI. If there is only one ESCI board in the DI, this parameter is optional. If there is more than one ESCI board in a DI, then the SLOT parameter is required to distinguish between ESCI boards.

TRUNK_NAME (TN)

Logical name of the Ethernet trunk. The default name is derived from the SLOT parameter, as in $ESCI3 and $ESCI4. The trunk name must be unique within the catenet.

MAX_FRAME_SIZE (MFS)

Maximum frame size the channel can transmit or receive. The default value is 1514 bytes. For the 1.2.5 release of CDCNET, this parameter value is fixed at 1514 bytes. Any other value will be ignored.

INTERFRAME_SPACING (IS)

Minimum time period in nanoseconds between sending of Ethernet frames after a transmission has completed. Default is 96 nanoseconds.

Responses Ether trunk <trunk_name> defined.

--ERROR-- The Device Interface does not contain an ETHERNET board.

--ERROR-- Board slot <slot_number> does not contain an ETHERNET board.

--ERROR-- The Device Interface contains more than one ETHERNET board--the slot must be specified.

--ERROR-- Device state not on.
Card slot = <slot_number>.

--ERROR-- Device already owned.
Card slot = <slot_number>.

--ERROR-- Trunk name <trunk_name> already defined.

--FATAL--Not enough memory currently exists for required table space.

Examples

```
define_ether_trunk trunk_name=ETHER1,slot=4

    Ether trunk ETHER1 defined.
```

DEFINE_HDLC_NET (DEFHN)

Purpose

Configures a CDCNET HDLC network solution using a previously defined HDLC trunk. An "unable to start" error leaves the network defined but not started.

Format

DEFINE_HDLC_NET
TRUNK_NAME = name
NETWORK_ID = 1..7FFFFFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
ROUTING_INFO_NETWORK = boolean
CONGESTED_THRESHOLD = 20..255
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..500000

Parameters

TRUNK_NAME (TN)

The logical name of the HDLC trunk to be used for the network solution. The HDLC trunk with this name must be configured prior to the execution of this command.

NETWORK_ID (NI)

The CDCNET network identifier of the HDLC network solution. The network ID must be unique within the catenet.

NETWORK_NAME (NN)

The logical name of the network solution used in subsequent commands referencing the network solution. The default name is constructed from the NETWORK_ID parameter, using the format *$NET_xxxxxxxx*, where *xxxxxxxx* is the network ID expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

The cost of the network solution. The cost of a network may be calculated by dividing 100 million by the data rate of the network in bits per second. Cost is used by CDCNET network routing to determine the least-cost routes to use to interconnect networks. For example, the cost of a trunk with a speed of 56,000 bits per second would be 06FA(16).

RELAY_ALLOWED (RA)

Indicates whether relay is allowed through this network solution. If RA is TRUE, then this network may be used as part of a route to interconnect two other networks. If RA is FALSE, then this network will be used only as part of an interconnecting route when no other route can be used to interconnect the networks. The default for an HDLC network is TRUE, relay allowed.

ROUTING_INFO_NETWORK (RIN)

Indicates whether or not the network solution is to carry CDCNET routing information. If RIN is TRUE, routing information describing all the networks to which this system is attached is sent over this network solution. If RIN is FALSE, routing information is not sent by this system over the network solution. This system would appear unconnected to any network other than this network solution. The default value is TRUE.

CONGESTED_THRESHOLD (CT)

For this release and future releases, this parameter is ignored.

START (S)

Specifies whether or not the configured element should be started. The default value is TRUE.

ARCHITECTURE_TYPE (AT)

Specifies the network architecture that this network supports. Allowed architecture types are CDNA and DOD. The DOD parameter value is currently not supported.

OUTPUT_QUEUE_LIMIT (OQL)

Specifies, in bytes, the maximum amount of data which is retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. The newest output messages are discarded first if messages need to be discarded.

The default value depends on the cost of the network (see COST parameter). If the cost is 6FA(16) or greater, then the default output queue limit is 30000 bytes. Otherwise, the default value is 60000 bytes.

Responses

HDLC network <network_name> defined for trunk <trunk_name>.

HDLC network <network_name> defined and started for trunk <trunk_name>.

--WARNING-- The value specified for the network_id, <value>, is greater than 65535 (0ffff(16)). Future CDCNET releases will not support a Network_id greater than 65535 (0ffff(16)).

--WARNING-- The 3A Command Processor timed out waiting for a response from the SSR. Please check network status for completion of requests.

--ERROR-- Network <network_name> already defined for trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not an HDLC trunk.

--ERROR-- Network name <network_name> already defined.

--ERROR-- Network id <network_id> already defined.

--FATAL-- Not enough memory is currently available for required table space.

--ERROR-- Trunk <trunk_name> down. Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> off.
Unable to start network <network_name>.

--FATAL-- Unable to start task <entry_point_name>.

--FATAL-- Stream Service error.

HDLC SSR received error when sending command to DVM.

--FATAL-- Stream Service error.

HDLC SSR received error on start port services.

Remarks

This command is only required in DIs that are to be configured with an HDLC network solution.

Examples

```
define_hdlc_net trunk_name=$lim5_port0,network_id=003(16) ..
network_name=hdlc_net3
```

DEFINE_HDLC_TRUNK (DEFHT)

Purpose Configures the layer 2 parameters of an HDLC network solution.

Format

DEFINE_HDLC_TRUNK
LIM = 0..7
PORT = 0..3
LOCAL_ADDRESS = 1..255
REMOTE_ADDRESS = 1..255
TRUNK_NAME = name
OPTIONS = list of 1..6 of keyword value
MAX_UNACK_FRAMES = 0..7
SREJ_QUEUE_SIZE = 0..7
MAX_FRAME_SIZE = 1500
PF_RECOVERY_TIMER = 500..65535
ERROR_RECOVERY_TIMER = 500..65535
RETRANSMISSION_LIMIT = 1..65535
TRUNK_SPEED = keyword value
CLOCKING = keyword value
INTERACTIVE_BANDWIDTH = 1..9

Parameters

LIM (L)

The LIM number for the port to which the HDLC line is connected.

PORT (P)

The port number for the port to which the HDLC line is connected.

LOCAL_ADDRESS (LA)

The address of the local HDLC station.

REMOTE_ADDRESS (RA)

The address of the remote HDLC station.

TRUNK_NAME (TN)

The logical name of the HDLC trunk. The default name will be constructed using the LIM and PORT parameters, as in $LIM1_PORT3.

OPTIONS (O)

Specifies the list of standard HDLC options to be supported by the trunk being configured. Allowed keyword values include the following:

Keyword Value	Description
REJ_ON	Includes a reject (REJ) code in the HDLC control field. REJ indicates detection of a tranmission error and requests retransmission of information frames.
REJ_OFF	Does not include a REJ code in the HDLC control field.
SREJ_ON	Includes a selective reject (SREJ) code in the HDLC control field. SREJ requests retransmission of only the information frame specified.
SREJ_OFF	Does not include a SREJ code in the HDLC control field.
UI_ON	Includes an unnumbered information (UI) code in the HDLC control field. UI transfers nonsequence-numbered information fields, such as higher level status and link initialization data, across a link. Reception of UI-labeled information frames is not verified by sequence number.
UI_OFF	Does not include a UI code in the HDLC control field.
SIM_ON	Includes a set initialization mode (SIM) code in the HDLC control field. SIM starts system-specific initialization procedures at the remote station.
SIM_OFF	Does not include a SIM code in the HDLC control field.
RESET_ON	Includes a reset code in the HDLC control field. Reset is transmitted by a combined station, and resets the receive state variable and frame reject (FRMR) condition in the addressed combined station. FRMR reports error conditions which cannot be recovered by retransmitting the frame in error. Error conditions may include a command that is not implemented or is invalid, an information field which exceeds maximum length, and an invalid receive sequence number.
RESET_OFF	Does not include a reset code in the HDLC control field.

Keyword Value	Description
IFRAME_ON	Includes an information frame (IFRAME) code in the HDLC control field. IFRAME transfers sequentially-numbered frames, including user information, across the data link. Counts are kept for the frame number being sent and the frame number expected to be next received. Each station continually reports these counts to each other during information exchange.
IFRAME_OFF	Does not include an IFRAME code in the HDLC control field.

The default list of HDLC options is (REJ_ON, SREJ_ON, UI_ON, SIM_ON, RESET_ON, IFRAME_ON).

MAX_UNACK_FRAMES (MUF)

The window size specifying the maximum number of frames the local station can send without receiving an acknowledgement. The value of this parameter can range from 0 through 7. The default value is 7.

SREJ_QUEUE_SIZE (SQS)

The size of the queue used to hold frames received out of sequence and being held by the HDLC SSR pending the receipt of missing frames whose transmission has been requested via the SREJ. The value of this parameter can range from 0 through 7. The default value is 7.

MAX_FRAME_SIZE (MFS)

The maximum frame size, in bytes, for the HDLC frame which may be transmitted or received. The value of this parameter can range from 2 through 65535. The default value is 1500 bytes. For the CDCNET 1.2.5 release, this value is fixed at 1500 bytes. All other values will be ignored.

PF_RECOVERY_TIMER (PRT)

The value of the timer in milliseconds. This timer is used to initiate the P/F recovery when an acknowledgement is not received for an IFRAME within this time period. The value of this timer can range from 500 through 65535. Its default value is 500.

ERROR_RECOVERY_TIMER (ERT)

The value of the error recovery timer in milliseconds. This is the timer used to determine if the P/F recovery has failed and to initiate the next level of recovery. The value of this timer can range from 500 through 65535. Its default value is 3000.

RETRANSMISSION_LIMIT (RL)

The maximum retransmissions allowed for a given control frame. The value of this parameter can range from 1 through 65535. The default value is 5.

TRUNK_SPEED (TS)

The speed of the HDLC trunk in bits per second. Trunk speed is used by the LIM to generate the data clocking for the trunk (except when clocking has been specified to be EXTERNAL), and to configure the media with the proper values for the network cost and output queue limit. The possible values for this parameter are:

1200
2400
4800
9600
19200
38400
48000
56000
64000

Default is 56000. Failure to specify this parameter for any speed other than 56000 bits per second will result in suboptimal performance.

CLOCKING (C)

Specifies whether the LIM internally generates the clock signal for data on this trunk or uses an externally generated clock signal for data on the trunk. If the LIM generates the data clock signal, the clocking rate is derived from the TRUNK_SPEED parameter. Allowed keyword values include EXTERNAL and TRANSMIT.

Keyword Value	Description
EXTERNAL	The LIM derives data clocking for both receive and transmit data from external signals (the TRUNK_SPEED parameter value is then informational, only). The EXTERNAL receive data clock is derived from the RS232 DD circuit for RS232 ports or the RS449 SR circuit for RS449 ports. The EXTERNAL transmit data clock is derived from the RS232 DB circuit for RS232 ports or the RS449 ST circuit for RS449 ports.
TRANSMIT	The LIM generates the clocking for transmit data, but derives the clocking for receive data from an external source. The transmit data clock matches the trunk speed specified for the line. The LIM supplies the transmit data clock on the RS232 DA or RS449 TT circuit. The LIM derives the receive data clock from the RS232 DD or the RS449 SR circuit.

Default clocking is EXTERNAL.

Clocking should be TRANSMIT for HDLC trunks connected directly between DIs (without intervening modems). Clocking should be EXTERNAL for HDLC trunks with modems.

In order for data clocking to work, make sure the LIMs supporting the HDLC trunks have the appropriate hardware configuration, in addition to setting the CLOCKING parameter on this command.

INTERACTIVE_BANDWIDTH (IB)

Specifies the percentage of the trunk bandwidth to be used to transmit data at interactive priority. The default value is 7. For example, a value of 7 on this parameter will, on average, result in 70 bytes of interactive priority data for every 30 bytes of batch priority data.

Responses

HDLC trunk <trunk_name> defined.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- Specified port value is greater than 1 for a 2 port LIM.

--ERROR-- LIM x, PORT y is not installed in this system.

--ERROR-- LIM xx, PORT xx addresses a port that cannot be serviced. More than 48 ports are attached to CIMxx. Ports beyond the 48th port attached to a CIM are not serviced.

--ERROR-- Not enough CIM memory available to load xxx I/O Processor.

--ERROR-- Specified LIM, PORT is already in use.

--FATAL-- Not enough memory is currently available for required table space.

--ERROR-- Specified LIM, PORT is not on.

--ERROR-- Line_speed <integer> is not supported for an HDLC trunk.

--ERROR-- HDLC is not supported on the specified LIM.

Remarks

This command is only required in DIs that are to be configured with an HDLC trunk.

Examples

```
define_hdlc_trunk lim=5,port=0,local_address=1,..
remote_address=26
```

DEFINE_IP_HOST (DEFIH)

Purpose

Configures an IP host address and associated static routing information, such as the Ethernet system ID. An IP host must be configured for the following:

- Every TCP/IP addressable host on a directly connected local area network with which this DI will be exchanging data.
- Every IP gateway that links any directly connected IP network to other IP networks that are not directly connected.
- Every IP address for which this DI provides services. That is, every CYBER that runs TCP/IP applications via this DI.

A DEFIH command is not required for:

- Any DI or host which does not use TCP/IP protocols.
- Any TCP/IP host on a directly connected wide area network, such as MILNET and ARPANET, where the physical address can be directly derived from the IP address.
- Any TCP/IP host that is not on a directly connected network. In other words, any host that can be reached only by traversing an intervening network.

Format

DEFINE_IP_HOST
IP_ADDRESS = list 4 of 0..255
HOST_TYPE = keyword value
SYSTEM_ID = list 2 of 0..0ffffff(16)
LAN_HEADER_FORMAT = keyword value

Parameters

IP_ADDRESS (IA)

The IP address of the host/workstation/PC to be configured. The network number portion of the IP address must have been previously defined by a DEFINE_IP_NET command. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

HOST_TYPE (HT)

Specifies the type of host associated with the configured address. The default value is IP_HOST. The following types are allowed:

Keyword Value	Description
LOCAL (L)	The specified IP address refers to the DI you are configuring: it is the default IP address for this DI. The IP address is the source address used by IP if upper layer protocols, such as TELNET, do not specify a source address. For Release 1.2.5, there can be only one IP address of host type LOCAL configured for each IP network to which a DI is physically connected.
CDC_HOST (CH)	The specified IP address refers to an alternate address for this DI. That is, it is another IP address for this DI, in addition to the IP address of the LOCAL host type. A DI can be configured to have several IP addresses. Of these addresses, only one address can be of type LOCAL for each connected IP network; all other addresses must be of type CDC_HOST. There should be a DEFIH with host_type of CDC_HOST for every CYBER that this DI is to access and/or service.
IP_HOST (IH)	The specified IP address refers to a host that is not the local DI. This host type is used to define other IP hosts on an Ethernet network that is directly connected to the DI, so that an IP address can be mapped to a physical Ethernet system ID, and vice versa. Other IP hosts need not be defined if their network is physically addressed by the IP address (for example, MILNET and ARPANET are physically addressed by the IP address), or if the IP network is not physically connected to the DI, but is reached via a gateway.
IP_GW (IG)	The specified IP address refers to a gateway that is not the local DI. This type is similar to the IP_HOST host type, except IP_GW further specifies that the host is a gateway, and can therefore be used as a route to another IP network. An IP address must be of type IP_GW if referenced in a DEFINE_IP_NET command.

SYSTEM_ID (SI)

Specifies the Ethernet address or CDNA system ID of the host. It is a 48-bit address, specified as a list of two 24-bit integers. For example, the Ethernet address 080025212345(16) is entered as (080025(16), 212345(16)). This parameter can be omitted for hosts that are not on Ethernet media or for HOST_TYPE CDC_HOST or LOCAL.

LAN_HEADER_FORMAT (LHF)

Specifies the type of local area network (LAN) header format that is used at the configured address. This parameter is ignored for HOST_TYPE CDC_HOST. The following types are allowed:

Keyword Value	Description
STANDARD_ HEADER (SH)	The host uses an IEEE 802.2 Ethernet header. This is the CDNA standard for CDCNET.
XNS_HEADER (XH)	The host uses a Xerox Networking Software (XNS) Version 2 Ethernet header. All older and most recent TCP/IP implementations use this header format.
IEEE_XNS_HEADER (IXH)	The host uses an IEEE 802.2 Ethernet header with a SNAP header to encapsulate XNS information. This is the new TCP/IP standard header format. However, most existing implementations use the older XNS_HEADER. Consult the vendor's documentation to determine which header is required.

Default is STANDARD_HEADER.

Responses

IP Host address <ip_address> is defined.

--ERROR-- IP Host address <ip_address> is already defined.

--ERROR-- IP Network <network_address> is not defined.

--ERROR-- IP Address <ip_address> is invalid.

--ERROR-- System id is required for host-type of IP_HOST and IP_GW.

--ERROR-- HOST_TYPE of LOCAL is already defined for IP Network <network_address>.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_ip_host ip_address = (128,5,0,3) ..
host_type = ip_host lan_header_format = xns_header ..
system_id = (020701(16),009ec9(16))

IP HOST address 128.5.0.3 is defined.
```

DEFINE_IP_NET (DEFIN)

Purpose

Configures an Internet Protocol network and associated routing information. An IP network address must be configured for every directly connected IP network. If the IP network is directly connected, the physical network must be previously defined by the DEFINE_ETHER_TRUNK and DEFINE_ETHER_NET commands.

Format

DEFINE_IP_NET
  IP_NETWORK = list 1..4 of 0..255
  IP_ADDRESS = list 4 of 0..255
  HOP_COUNT = 0..255
  MAX_DATAGRAM_SIZE = 20..1518
  TRUNK_NAME = name

Parameters

IP_NETWORK (IN)

Specifies the IP network number portion of the IP address of the network to be configured. If this parameter is set to zero, any datagrams to IP networks that are not in the routing tables are sent to the host specified by the IP_ADDRESS parameter. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP network 192.2.53.0 is represented as (192,2,53,0) or (192,2,53).

IP_ADDRESS (IA)

The IP address of the next gateway (hop) in the route to the destination IP network. This host must subsequently be configured by a DEFINE_IP_HOST command for the network to actually be reached. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

HOP_COUNT (HC)

Specifies the number of hops, or gateways, that must be traversed to reach this configured IP network. If the hop count is zero, the network is a directly connected network and the TRUNK_NAME parameter (see below) must also be specified. The default is 0.

MAX_DATAGRAM_SIZE (MDS)

Specifies the maximum datagram size (in bytes) that the IP network can handle without fragmentation. If the hop count is nonzero, the maximum size of intervening IP networks should also be considered to avoid fragmentation. The default is 576 bytes.

TRUNK_NAME (TN)

Specifies the CDCNET trunk name of a directly connected network. The name must have been previously specified using the DEFINE_ETHER_TRUNK or the DEFINE_X25_TRUNK command (X.25 support for TCP/IP is not supported for CDCNET 1.2.5). This parameter is required if the value of the HOP_COUNT parameter is zero, and is ignored otherwise.

Responses

IP Network <ip_network> is defined.

--ERROR-- IP Network <ip_network> is already defined.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk name must be specified if hop_count is zero.

--ERROR-- IP Address <ip_address> is invalid.

--ERROR-- IP Address <ip_address> is not defined.

--ERROR-- IP Network <ip_network> is invalid. Only the network number part of an IP address should be specified.

--ERROR-- IP Address must be specified if hop_count is nonzero.

--ERROR-- IP Address <ip_address> must be on a directly connected IP network.

--ERROR-- Trunk id is not configured. A DEFINE_ETHER_TRUNK command is required.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_ip_net ip_network=(1,0,0,0) trunk_name = $net_1 ..
ip_address = (128,5,0,0) hop_count = 0 ..
maximum_datagram_size = 576

IP Network 1.0.0.0 is defined.
```

DEFINE_LINE (DEFL)

Purpose

Defines a single terminal communication line or URI parallel interface line in terms of the logical line name, physical hardware address, name of TIP that services the line, physical line attributes, and connect timeout values.

The examples at the end of this description show DEFINE_LINE being entered in two ways: in a configuration procedure and through the Network Operator Utility (NETOU) while the network is running.

Format

DEFINE_LINE
LIM = 0..7
PORT = 0..7
TIP_NAME = name
LINE_NAME = name
LINE_TYPE = keyword value
LINE_SUB_TYPE = name
CARRIER_TYPE = keyword value
LINE_SPEED = keyword value
AUTO_RECOGNITION = keyword value
TRANSMISSION_BLOCK_SIZE = 128..4095
CONNECTION_CONNECT_TIMEOUT = 20..1000 or keyword value
CONNECTION_DISCONNECT_TIMEOUT = 0..1000 or keyword value
TERMINAL_DEFINITION_PROCEDURE = name
TERMINAL_USER_PROCEDURE = name
START = boolean
USER_CONNECTION_LIMIT = 1..16
EIA_FLOW_CONTROL = boolean
CLOCKING = keyword value
DATA_PARITY= keyword value

Parameters

LIM (L)

Specifies the slot number for the Line Interface Module (LIM) or Unit Record Interface (URI) board in the MTI/TDI to which the line is connected. An MTI or TDI allows for up to 8 LIMs/URIs to be installed, which determines that the range of this parameter is 0 through 7.

PORT (P)

Specifies the LIM port number that connects to the line. The number of ports supported per LIM is LIM model-specific. Depending on the LIM model supporting the line, the range for this parameter may be 0 through 1, 0 through 3, or 0 through 7.

TIP_NAME (TN)

Defines the type of TIP that services the line. Refer to the DEFINE_TIP command (parameter TIP_NAME or USER_TIP_NAME for user-defined TIPs) for a definition of allowed TIP types.

If the line is being configured for XPCTIP usage, two methods of specifying the TIP are available:

- Specify ASYNCTIP if the X.PC user is required to first connect as an asynchronous ASCII terminal before switching to the XPCTIP. This allows the line to service both asynchronous ASCII terminals and X.PC personal computers. The switch to the XPCTIP is accomplished from the personal computer with the terminal user command, ACTIVATE_X_PERSONAL_COMPUTER (ACTXPC). Make sure that the XPCTIP is configured by a DEFINE_TIP command before any users attempt to switch from the ASYNCTIP to the XPCTIP. Otherwise, users will receive an error message indicating that X.PC is not defined for their line when they enter the ACTXPC command.

- Specify XPCTIP if the X.PC user can connect directly to the XPCTIP without first connecting to the ASYNCTIP.

NOTE

For the XPCTIP, if terminal definition procedures (TDPs) are used to configure the terminal devices for an X.PC line (reference DEFINE_TERMINAL_DEVICE command), the TDP must not contain a DEFTD command with a non-zero device address. The X.PC protocol will start only when the device address is set or defaulted to zero.

LINE_NAME (LN)

Specifies the logical name of the line or URI parallel interface. The default line name is constructed from the values of the LIM and PORT or URI parameters, as in $LIM3_PORT1 or $URI2.

If the TIP_NAME is ASYNCTIP, and the DI is connected to a NOS host, the NOS terminal name will be based on the line name (unless a TDP containing a DEFINE_TERMINAL_DEVICE command that names the terminal is also specified). It is a site administrator's responsibility to ensure that the terminal name for each line is unique throughout the network.

LINE_TYPE (LT)

Defines the type of line. SWITCHED and DEDICATED are the types allowed. The default is SWITCHED.

When defining a communication line that will use the URI TIP to support a 585 printer, the line should be defined as DEDICATED. If the line is defined as SWITCHED, or if the LINE_TYPE parameter is omitted (and the LINE_TYPE parameter defaults to SWITCHED) several problems could occur:

- If there are periods of inactivity between printing files, the printer line will occasionally be automatically stopped and restarted. This will cause any batch device attributes made since the last time the line was started (by the CHANGE_BATCH_DEVICE_ATTRIBUTES command) to be lost.

- If the printer is powered off, there could be an extremely large number of messages sent to the log file or sent as alarms to the network operator.

With a dedicated URI printer line, a power-off condition will still cause log messages to occur, but with less frequency than a switched line (about every two minutes). If a printer is powered off for long periods of time, it is recommended that you stop its line with a STOP_LINE command (see CDCNET Network Operations manual).

LINE_SUB_TYPE (LST)

Defines the subtype of line. The subtype can be used by a site to further qualify the line type, such as WATS or INWATS. This parameter has no effect for this release of CDCNET.

CARRIER_TYPE (CT)

Defines the type of carrier control on the line. CONSTANT and CONTROLLED are the keyword values allowed. Default is CONSTANT. This parameter is ignored if TIP_NAME = URITIP.

LINE_SPEED (LS)

Defines the line speed of a communication line in bits per second. The following line speeds are allowed.

50
75
110
150
300
600
1200
1800
2400
3600
4800
7200
9600
19200
38400
48000
56000
64000

Default is 1200 (if not requesting auto recognition of speed). The range of line speeds from 50 through 38400 is the range supported by the asynchronous and X.PC TIPs. The range of line speeds from 1200 through 64000 is supported by the HASP, BSC3270, BSCNJEF, and NTF TIPs. The MODE4 TIP supports a range from 1220 through 19200.

This parameter is informational only, if the LIM board does not generate the data clocking for the line or if auto recognition of the line speed is requested for an asynchronous line.

This parameter is ignored if TIP_NAME = URITIP.

AUTO_RECOGNITION (AR)

Defines the type of auto recognition to be performed for asynchronous lines. Allowed values include the following:

Keyword Value	Description
NONE	No auto recognition. Default value.
S	Auto recognition of line speed only.
SC	Auto recognition of line speed and code set.
SCP	Auto recognition of line speed, code set, and parity. The only parity types recognized are odd and even.

If auto recognition of code set and parity is not requested, ASCII code set and even parity are assumed. A terminal user can change these values by the CHANGE_TERMINAL_ATTRIBUTES command. On a switched line, a terminal user has 90 seconds to complete the auto recognition logic.

This parameter is ignored if TIP_NAME = URITIP.

X.PC line speed can be automatically recognized; however, character set and parity are ignored. The X.PC character set is always ASCII and parity is set by the X.PC protocol. Please refer to the X.PC terminal attribute, PARITY, described in appendix H of the CDCNET Terminal Interface manual.

TRANSMISSION_BLOCK_SIZE (TBS)

Defines the transmission block size to be used for transmission blocks exchanged with the terminal device(s) on this line. Values range from 128 (80(16)) through 4095 (0FFF(16)). The value may be specified in hexadecimal form. Default value is the TBS on the DEFINE_TIP command. This parameter applies to the following TIPs: HASP, MODE4, BSCNJEF, BSC3270, NTF.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines the amount of time the line user has to create the first $input/$output connection. If no connection is established within that time, the line is disconnected. The range is 20 through 1000 seconds, and the keyword value INFINITE. The default is 120 for a switched line and INFINITE for a dedicated line. The keyword value INFINITE indicates an infinite time. This parameter is rounded up to the nearest multiple of 4 seconds. As a result, there may be a discrepancy between the value specified on this command and the value displayed in the response to the DISPLAY_LINE_OPTIONS command (entered through NETOU) for this line. This timeout value does not include the possible auto recognition time. This parameter is ignored if TIP_NAME = URITIP.

CONNECTION_DISCONNECT_TIMEOUT (CDT)

Defines the amount of time the line user has to establish a new $input/$output connection after the last such connection has been disconnected. If no new connection is established within that time, the line will be disabled and reenabled, causing a switched line to be disconnected or the modem signals of a hardwired line to be dropped for a period of time. The range is 20 through 1000 seconds, and the keyword value INFINITE. The default is 120 for a switched line and INFINITE for a dedicated line. The keyword value INFINITE indicates an infinite time. This parameter is rounded up to the nearest multiple of 4 seconds. As a result, there may be a discrepancy between the value specified on this command and the value displayed in the response to the DISPLAY_LINE_OPTIONS command (entered through NETOU) for this line. This timeout value does not include the possible auto recognition time.

TERMINAL_DEFINITION_PROCEDURE (TDP)

Name of a terminal definition procedure (TDP) file. The commands within the named file are executed when the line becomes active. If both a TUP and TDP parameter are specified on a DEFINE_LINE command, the TUP parameter is ignored and only the TDP is executed. You can specify a TDP on a DEFINE_LINE command if you want to define a terminal device in a way that differs from the defaults set by the TIP controlling the lines and terminal devices connected to the DI.

TERMINAL_USER_PROCEDURE (TUP)

Specifies the name of a terminal user procedure (TUP) file to be executed when the DEFINE_LINE command executes. The commands within the TUP specified by this parameter are executed for each interactive device on the line that becomes active.

The TUP parameter is ignored if the TERMINAL_DEFINITION_PROCEDURE (TDP) parameter is also used to specify a TDP to be executed for this line. A TUP is not executed if a TDP parameter is specified on a DEFINE_LINE command. If you use the TDP parameter but you also want to use a TUP to define terminals on this line, the commands in the TDP used for the line must specify any TUPs to be executed for the line.

This parameter is ignored if TIP_NAME = URITIP.

You can specify a TUP on a DEFINE_LINE command if you want to define a terminal's characteristics in a way that differs from the defaults set by the TIP controlling the lines and terminal devices connected to the DI. The TUP name specified here overrides the value of the TUP parameter on the DEFINE_TIP command.

START (S)

Specifies whether or not the line should be started after it is configured. Possible values are TRUE, start; and FALSE, do not start. Default is TRUE.

USER_CONNECTION_LIMIT (UCL)

Defines the maximum number of connections that a user of the line can have outstanding at one time. This maximum number of connections may range from 1 through 16. Default is 4 connections. This parameter is ignored if TIP_NAME = URITIP.

For lines supported by the XPC TIP, user connections are counted in the following manner: each virtual circuit without an $input/$output connection is counted as one connection. All other virtual circuits are counted as equal to the number of $input/$output connections they have. When the user connection limit is reached, no new virtual circuits or $input/$output connections are permitted.

EIA_FLOW_CONTROL (EFC)

Specifies whether the Clear to Send and Request to Send flow control will be used to stop and resume the flow of input and output data. The options are ON and OFF. Default is OFF. This parameter is ignored if TIP_NAME = URITIP. The LIM cables must support flow control if this parameter is set to ON.

CLOCKING (C)

Specifies whether the LIM internally generates the clock signal for data on this line or uses an externally-generated clock signal. If the LIM generates the data clock signal, the clocking rate is derived from the LINE_SPEED parameter. This parameter is ignored for URI lines. The following keyword values are allowed.

Keyword Value	Description
EXTERNAL	Specifies that the LIM derives data clocking for both receive and transmit data from external signals (LINE_SPEED is then informational only). The EXTERNAL receive data clock is derived from the RS232 DD circuit for RS232 ports or the RS449 SR circuit for RS449 ports. The EXTERNAL transmit data clock is derived from the RS232 DB circuit for RS232 ports or the RS449 ST circuit for RS449 ports.
INTERNAL	Specifies that the LIM generates the required clocking signals for both transmit and receive data (with NULL modem cable TN109). A single clock signal is generated; it matches the line speed specified for the line. The LIM supplies the clock on the RS232 DA or RS449 TT circuit.
TRANSMIT	Specifies that the LIM generates the clocking for transmit data but derives the clocking for receive data from an external source. The transmit data clock matches the line speed specified for the line. The LIM supplies the transmit data clock on the RS232 DA or RS449 TT circuit. The LIM derives the receive data clock from the RS232 DD or the RS449 SR circuit.

Default clocking is INTERNAL.

Clocking should be set to INTERNAL for asynchronous communication lines. For synchronous terminals that provide the transmit clock, set CLOCKING to TRANSMIT (with NULL modem cable TN109). Most terminals will generate the transmit clock as defined by the RS232 standard. When using a modem, CLOCKING must be set to EXTERNAL and modem cable TN108 must be used, since the modem will generate both clocking signals. Inappropriate selection of INTERNAL clocking can cause data to be received with errors or not received at all.

DATA_PARITY (DP)

Specifies parity for data received and transmitted on a line. The following keyword values are allowed.

Keyword Value	Description
ZERO	The parity bit is always zero.
MARK	The parity bit is always 1.
EVEN	The parity bit is set so that the sum of the parity and data bits is an even value.
ODD	The parity bit is set so that the sum of the parity and data bits is an odd value.
NONE	The parity bit is considered a data bit.

Default data parity is EVEN. This parameter is ignored by the URITIP. For the XPCTIP, the allowed values are ZERO and NONE. Parity type of NONE is significant only during transparent input or output. In all other cases, parity is treated as ZERO.

Responses

Line <name> defined.

--ERROR-- LIM <xx>, port <yy> addresses a port that cannot be serviced. More than 48 ports are attached to CIMxx. Ports beyond the 48th port attached to a CIM are not serviced.

--ERROR-- Not enough CIM memory available to load <xxx> I/O processor.

--ERROR-- LIM <xx>, port <yy> not responding or not installed.

--ERROR-- TIP name <name> is not a CDC defined TIP name.

--ERROR-- TIP <xxx> is not defined.

--ERROR-- TIP type <xxx> is not supported on the 8-port LIM.

--ERROR-- Line <name> previously defined.

--ERROR-- Line speed <line_speed> is not supported for the specified TIP.

--ERROR-- Load module <TIP name>_CIM is not available.

--ERROR-- LIM <xx>, port <yy> not defined, hardware status indicates port is in a DOWN or OFF state.

--ERROR-- LIM <xx>, port <yy> not defined, LIM and port previously defined.

--ERROR-- 3270 TIP will not operate with LIM <xx>.

--ERROR-- Unable to define TIP <TIP name>. No CIM is installed.

--FATAL-- Not enough memory is currently available for required table space.

Remarks

Two timers on lines cannot be configured and are assigned fixed values by terminal support software. These timers are the delay reenable for switched lines and delay reenable for dedicated lines. They control the action taken when a line disconnects and the amount of time that elapses until the line can be reenabled. For switched lines, the time is 2 seconds after the line disconnects. For dedicated lines, this timeout varies according to the TIP supporting the line. For ASYNCTIP and XPCTIP-supported lines the time is 5 seconds after the line disconnects. For all other TIPs, the delay reenable for dedicated lines is 2 minutes. During this time, a line user cannot perform auto recognition or connect to CDCNET.

Examples

The following example shows a DEFINE_LINE command as it would be entered in a configuration procedure for a TDI or MTI. This example defines a synchronous line with a line speed of 9600 bits per second that is controlled by the HASP TIP. The equipment connected to the line is further defined by the commands in a TDP named STATION2.

```
define_line line_name=line11,lim=1,port=1,tip_name=hasptip,..
line_type=dedicated,line_speed=9600,..
terminal_definition_proceduredp=station2
```

The following example shows a DEFINE_LINE command being entered using the Network Operator Utility (NETOU) to define an asynchronous line while a network is running. A network operator uses the NETOU SEND_COMMAND to send the DEFINE_LINE command to the TDI connected to the line. NETOU is invoked by entering the NETWORK_OPERATOR_UTILITY (NETOU) command on NOS/VE and by selecting the NETOU application during the login process on NOS. The nou/ in the example is a prompt sent by NETOU when NETOU is currently invoked. The DEFINE_LINE command itself is sent as a string value within SEND_COMMAND.

Unlike the first example, where the line was defined through a command in a configuration procedure, this example shows a temporary configuration change. That is, when the TDI resets and its software is reloaded, the line defined in this example will not be redefined. In order for the line to be redefined every time the TDI's software is reloaded, this DEFINE_LINE command would have to be placed in the TDI's configuration procedure. (For more information about making configuration changes while the network is running, refer to the Network Control chapter of the CDCNET Network Operations manual.)

```
nou/send_command command='define_line line_name=line23,..
lim=2,port=3,tip_name=asynctip,line_speed=9600',system=tdi_84
```

DEFINE_PASSTHROUGH_SERVICE (DEFPS)

Purpose Installs the Interactive Passthrough Gateway (IPG) application and optionally selects a passthrough connection timeout value. This command should be present in the configuration files of all DIs that have passthrough ports connected to them.

Format **DEFINE_PASSTHROUGH_SERVICE**
TITLE = name
INACTIVITY_TIMER = 120..14400 or keyword value
START = boolean

Parameters *TITLE (T)*

The title of the passthrough service. Passthrough ports are connected to the passthrough service using a CREATE_CONNECTION command with a SERVICE_NAME parameter equal to the value of this parameter. The default value is PASSTHROUGH.

INACTIVITY_TIMER (IT)

The maximum time in seconds that a passthrough connection can remain idle. Idle means that no data has been transferred in either direction on the connection. When this timer value is exceeded, the passthrough connection to the terminal user is disconnected. The keyword value INFINITE specifies that passthrough connections are not to be timed out. The default value is INFINITE.

START (S)

Specifies whether or not the defined service is to be started. The default value is YES.

Responses Passthrough Service <title> defined.

Passthrough Service <title> defined and started.

--WARNING-- Passthrough Service <title> defined but not started.

--ERROR-- Passthrough Service previously defined.

--FATAL-- Not enough memory is currently available for required table space.

Remarks See also the DEFINE_PASSTHROUGH_TITLES command in the CDCNET Terminal Interface Usage manual.

Examples This example shows a passthrough service being defined and started. The title of the passthrough service in this example is different from the default title.

```
define_passthrough_service title=termpass
```

DEFINE_PRINTER_MODEL_ATTRIBUTES (DEFPMA)

Purpose Defines the printer attributes to be set for a specific printer terminal model. The printer terminal model defined with this command can be referenced on a DEFINE_BATCH_DEVICE command to specify that the batch device is a particular printer model, having that model's attributes. The DEFPMA command can be specified only in a DI configuration file.

Format **DEFINE_PRINTER_MODEL_ATTRIBUTES**
TERMINAL_MODEL = name
AUTO_PAGE_EJECT_CHANNEL = 2..12
CHANNEL_1_SEQUENCE = list 1..7 of <ccode>
CHANNEL_2_SEQUENCE = list 1..7 of <ccode>
CHANNEL_3_SEQUENCE = list 1..7 of <ccode>
CHANNEL_4_SEQUENCE = list 1..7 of <ccode>
CHANNEL_5_SEQUENCE = list 1..7 of <ccode>
CHANNEL_6_SEQUENCE = list 1..7 of <ccode>
CHANNEL_7_SEQUENCE = list 1..7 of <ccode>
CHANNEL_8_SEQUENCE = list 1..7 of <ccode>
CHANNEL_9_SEQUENCE = list 1..7 of <ccode>
CHANNEL_10_SEQUENCE = list 1..7 of <ccode>
CHANNEL_11_SEQUENCE = list 1..7 of <ccode>
CHANNEL_12_SEQUENCE = list 1..7 of <ccode>
FORM_FEED_DELAY = 0..3000
FOLD_LINE = boolean
FORM_FEED_SEQUENCE = list 1..7 of <ccode>
KEYBOARD = boolean
NO_SPACE_SEQUENCE = list 1..7 of <ccode>
SINGLE_SPACE_DELAY = 0..1000
SINGLE_SPACE_SEQUENCE = list 1..7 of <ccode>
BOTTOM_OF_FORM_CHANNEL = 1..12
VFU_TOP_FORM = boolean
MAXIMUM_VFU_LENGTH = 0..255

Parameters **TERMINAL_MODEL (TM)**

The name of the printer terminal model (1- through 25-characters long) for which attributes are being defined. This parameter may not be the same as any terminal model already defined by Control Data or by your site.

AUTO_PAGE_EJECT_CHANNEL (APEC)

This parameter is supported by the URI TIP only. Defines the channel the printer recognizes as causing it to skip automatically to the next top-of-form channel. The value for this parameter may not be the same as the value for BOTTOM_OF_FORM_CHANNEL. The default value is 2.

CHANNEL_1_SEQUENCE (C1S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever an "8" or "H" format effector is recognized in output lines. No default.

CHANNEL_2_SEQUENCE (C2S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "7" or "G" format effector is recognized in output lines. No default.

CHANNEL_3_SEQUENCE (C3S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "6" or "F" format effector is recognized in output lines. No default.

CHANNEL_4_SEQUENCE (C4S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "5" or "E" format effector is recognized in output lines. No default.

CHANNEL_5_SEQUENCE (C5S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "4" or "D" format effector is recognized in output lines. No default.

CHANNEL_6_SEQUENCE (C6S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "3" or "C" format effector is recognized in output lines. No default.

CHANNEL_7_SEQUENCE (C7S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "9" or "I" format effector is recognized in output lines. No default.

CHANNEL_8_SEQUENCE (C8S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "X" or "J" format effector is recognized in output lines. No default.

CHANNEL_9_SEQUENCE (C9S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "Y" or "K" format effector is recognized in output lines. No default.

CHANNEL_10_SEQUENCE (C10S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "Z" or "L" format effector is recognized in output lines. No default.

CHANNEL_11_SEQUENCE (C11S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "W" or "M" format effector is recognized in output lines. No default.

CHANNEL_12_SEQUENCE (C12S)

This parameter is currently not supported. Defines the sequence of 0 through 7 octets sent to the printer whenever a "U" or "N" format effector is recognized in output lines. No default.

FORM_FEED_DELAY (FFD)

This parameter is currently not supported. Defines the number of milliseconds (maximum 3000) that the TIP needs to delay after sending a CHANNEL_x_SEQUENCE value to the printer. The default value is 1000 milliseconds.

FOLD_LINE (FL)

This parameter is supported by the Asynchronous and URI TIPs only. Indicates if the TIP must fold lines that are longer than the page width of the device. The default value is YES.

FORM_FEED_SEQUENCE (FFS)

This parameter is currently not supported. Defines the sequence of octets sent to the printer when a "1" or an "A" format effector is recognized in output lines. The default value is 0C(16) (FF).

KEYBOARD (K)

This parameter is currently not supported. A boolean value that indicates if the printer has an associated keyboard. The default value is NO.

NO_SPACE_SEQUENCE (NSS)

This parameter is currently not supported. Defines the sequence of octets sent to the printer whenever a "+" format effector is recognized in output lines. The default value is 0D(16) (CR).

SINGLE_SPACE_DELAY (SSD)

This parameter is currently not supported. Defines the number of milliseconds (maximum 1000) that the TIP needs to delay after sending a SINGLE_SPACE_SEQUENCE to the printer. The default value is 50.

SINGLE_SPACE_SEQUENCE (SSS)

This parameter is currently not supported. Defines the sequence of octets sent to the printer whenever a blank (" "), a zero ("0"), or a hyphen ("-") format effector is recognized in output lines. The default value is 0D0A(16) (CR, LF).

BOTTOM_OF_FORM_CHANNEL (BOFC)

This parameter is supported by the HASP and URI TIPs. It is currently not supported by the ASYNC TIP. Specifies the channel to which the printer is to skip when a "2" or "B" format effector is recognized in output lines. The value for this parameter may not be the same as the value for AUTO_PAGE_EJECT_CHANNEL. The default value is 6.

VFU_TOP_FORM (VTF)

This parameter is currently supported by the URI TIP only. Defines whether the printer needs to be at top-of-form when the vertical format unit (VFU) load image is loaded. The default value is YES.

MAXIMUM_VFU_LENGTH (MVL)

This parameter is supported by the URI TIP only. Defines the maximum number of lines the printer supports in a vertical format unit (VFU) load image. The default value is 127 lines.

For a printer defined to support VFU loading (that is, if the VFU_LOAD_OPTION parameter on the DEFINE_BATCH_DEVICE command has any other value than NONE), it is important that the MVL parameter value does not exceed the maximum VFU length actually supported by the printer. If the MVL parameter value exceeds the actual supported length, attempts to load a VFU load image into the printer could fail. No files will be sent to the printer until the problem is corrected.

Responses

Printer model <xxx> defined.

--ERROR-- Printer model <xxx> is already defined.

--ERROR-- The bottom of form and auto page eject channels cannot be the same.

Remarks

You can use the DEFINE_PRINTER_MODEL_ATTRIBUTES command when specifying printer attributes for a non-Control Data printer, or when you want to change the default printer attributes Control Data provides for various terminal models. Note that when you define new printer models, the terminal model name must be unique. You cannot give the printer the same name as one of the default terminal model names, such as C585V or C18.

Use the DEFINE_PRINTER_MODEL_ATTRIBUTES command only in the system configuration procedures of DIs having batch devices that will use the parameter values set by the command. Since an MDI does not have any batch devices, it does not need to have a DEFINE_PRINTER_MODEL_ATTRIBUTES command.

Examples

The following example redefines a Control Data 585 printer to have an auto page eject channel of 11 (the default auto page-eject channel for the Control Data 585 printer is 8).

```
define_printer_model_attributes terminal_model=user_585 ..
auto_page_eject_channel=11 vfu_top_form=yes ..
bottom_of_form_channel=12
```

The next example defines a printer model with no vertical format unit (VFU). The printer has a bottom-of-form channel of 11. The printer itself will do line folding.

```
define_printer_model_attributes terminal_model=non_585a ..
    bottom_of_form_channel=11 fold_line=no vfu_top_form=no ..
    maximum_vfu_length=0
```

This next example defines a printer model with a VFU. The printer has an auto page eject channel of 2, and a bottom-of-form channel of 12. The printer does not support line folding. It can support a maximum VFU size of 256 lines.

```
define_printer_model_attributes terminal_model=non_585b ..
    bottom_of_form_channel=12 maximum_vfu_length=256
```

DEFINE_REMOTE_LOAD_SUPPORT (DEFRLS)

Purpose Defines and starts the remote load support network management service (Independent Initialization Management Entity) in a DI. When remote load support is defined in a DI, it can load other DIs over a network to which both DIs are directly connected.

Format **DEFINE_REMOTE_LOAD_SUPPORT**
PRIORITY = 0..3
CONCURRENT_LOAD_LIMIT = 0..8
RESTRICTED_NETWORK = list 1..15 of name

Parameters *PRIORITY (P)*

Specifies the priority of the "help offer" that a DI containing remote load support sends to remote systems when they request to be loaded. The default value for this parameter, and the highest priority value, is 3.

The DI to be loaded uses the help offer's priority to decide if it should accept the help offer. The DI to be loaded accepts a help offer right away if its priority is 3. However, if the priority of the help offer is less than 3, the DI to be loaded waits for a certain period before it accepts a help offer. During this period, if the DI to be loaded receives a help offer at priority 3, it accepts that help offer. Otherwise, at the end of this period, it selects the highest priority help offer among all help offers received during this period.

You can use this parameter to assign backup remote load support to a DI. For example, you can assign one DI to provide primary remote load support by using the default value for this parameter. You can assign backup remote load support to another DI by defining remote load support and assigning it a lower priority, such as 2. If the first DI cannot respond to load requests, the second DI will.

CONCURRENT_LOAD_LIMIT (CLL)

Specifies the maximum number of DIs which may be simultaneously loaded by the DI providing remote load support. The default value for this parameter is 4. You can use this parameter to prevent a DI from loading more DIs than the limit you set. When the number of DIs being concurrently loaded equals the limit set, the DI will not respond to load requests from other DIs.

RESTRICTED_NETWORK (RN)

Specifies the names of networks over which a DI containing remote load support should not load other DIs. When this parameter is specified, the remote load support in a DI will not respond to load requests from DIs that are on the restricted network or networks. The default value for this parameter is an empty list; by default, a DI containing remote load support will load remote DIs over all directly connected network solutions.

Responses Remote Load Support is defined.

--ERROR-- Remote Load Support is already defined.

--FATAL-- Remote Load Support can not be defined at this time. Not enough memory is currently available for required table space.

Examples

```
define_remote_load_support priority=1 ..
restricted_network=hdlc_net
```

DEFINE_SERVER_TELNET_GW (DEFSTG)

Purpose

Configures a server TELNET gateway, which provides access to the interactive terminal services of a CYBER host to remote terminal users on hosts connected via a TCP/IP network.

If both terminal (via TELNET) and application (via FTP) services are to be provided for the same IP address, the application gateway (TCP/IP gateway) must be defined in the same DI as the server TELNET gateway. It is not possible for more than one DI to service the same IP address.

The timeout parameters: TCP_TIMEOUT for TCP and INACTIVITY_TIMEOUT for TELNET impose no limits on the user. That is, a user can leave a connection idle for any period of time without losing the connection. Note that the host service may impose inactivity limits.

Format

DEFINE_SERVER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = keyword value
MAX_CONNECTIONS = 0..65535 or keyword value
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..7fffffff(16)
TCP_TIMEOUT = 0..65535 or keyword value
INACTIVITY_TIMEOUT = 0..65535
START = boolean

Parameters

GATEWAY_NAME (GN)

The logical name of the server TELNET gateway used in subsequent commands that reference the gateway.

IP_ADDRESS (IA)

The IP address of the host for which this gateway provides server TELNET terminal service. The format is similar to the decimal octet convention used by the TCP/IP community, except that the periods are replaced with commas, and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

TITLE (T)

Specifies the title that this gateway translates to locate the service provider. If the destination system is NOS, this title must be from the DEFINE_NP_TERMINAL_GW command. If the destination system is NOS/VE, this title must be the one registered by the terminal manager. The default value is supplied from the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the CDCNET catenet that should be searched for the service corresponding to the title information given in the TITLE parameter. For CDCNET Release 1.2.5, the only supported value is CATENET. The default value is CATENET.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous connections to be supported by the gateway. If INFINITE is entered, there is no restriction to the number of connections allowed. The default value is INFINITE.

TCP_PORT_NUMBER (TPN)

Specifies the TCP port number to be used by the gateway. If omitted, the default is the well-known server TELNET port 23. Server TELNET issues a TCP passive_connect request using the well-known port for the source port.

TCP_ALLOCATE_SIZE (TAS)

Specifies the amount of data that the gateway queues for each connection. Larger values may improve user response time, especially for PC users (with a standard protocol such as XMODEM) but can increase the number of instances of DI congestion. Changing this value is discouraged, and should be done with caution, as network service may be disrupted. The default value is 4096 bytes.

TCP_TIMEOUT (TT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgment of data transmission. If an acknowledgment is not received within the specified period, TCP aborts the connection. A small value (less than a few seconds) may cause frequent and unnecessary loss of service during periods of network congestion. A large value may leave users waiting a long period of time after a host or network has failed. If INFINITE is entered, the connection does not timeout. The default value is 300 seconds.

INACTIVITY_TIMEOUT (IT)

Specifies the interval between inactivity checks, in seconds. If a connection has been idle for the specified time, the gateway sends a TELNET status request to the remote TELNET to determine if the connection is still usable. The default is 600 seconds.

START (S)

Specifies that the newly configured gateway is to be started after it is defined. The default value is TRUE.

Responses

Server TELNET gateway <gateway_name> is defined and started.

Server TELNET gateway <gateway_name> is defined.

--ERROR-- Server TELNET gateway <gateway_name> is already defined.

--ERROR-- IP Address <ip_address> is not defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_server_telnet_gw gateway_name = gw_to_cyber ..
ip_address = (128,5,0,2) title = ve106

Server TELNET gateway GW_TO_CYBER is defined and started.
```

DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM)

Purpose Defines the alarm messages (by specifying alarm message numbers) that the DI should send to the network operator. If this command is not used to configure a DI, no alarms will be generated by the DI.

Format **DEFINE_SOURCE_ALARM_MESSAGE**
MESSAGE_NUMBER = list 1..63 of integer 1..32999

Parameters **MESSAGE_NUMBER (MN)**

List of message numbers the DI is to send as alarms to the network operator. If this parameter is omitted, a set of default alarm message numbers are enabled. Refer to appendix D, Default Log and Alarm Messages, of the CDCNET Configuration and Site Administration Guide, for the alarm message numbers and their message identifiers. You may add alarms to this list using additional DEFSAM commands. You may also cancel messages using the CANCEL_SOURCE_ALARM_MESSAGE command (see CDCNET Network Operations manual); however, cancelling any of the default alarms is not recommended. For the complete list of diagnostic messages, refer to the CDCNET Diagnostic Messages manual.

Responses Source alarm messages defined.

--ERROR-- Source alarm messages are already defined.

--FATAL-- Not enough memory currently exists for required table space.

Remarks If more than one DEFSAM command is issued to a DI, the set of alarm messages defined for the DI is the set specified on the most recent occurrence of the command, in addition to any messages specified on any previous DEFSAM commands (including the default alarm message numbers).

For this release of CDCNET, the DEFSAM command automatically defines the alarm group CATENET.

Examples

```
define_source_alarm_message
```

DEFINE_SOURCE_LOG_GROUP (DEFSLG)

Purpose Defines the types of log messages to be logged by this DI, and defines the log groups to which this DI belongs. If this command is not used to configure a DI, no messages will be logged by the DI.

Format **DEFINE_SOURCE_LOG_GROUP**
LOG_GROUP = list 1..1 of name
MESSAGE_NUMBER = list 1..63 of integer 1..32999

Parameters *LOG_GROUP (LG)*

Name of the source log group to which the Dependent Log ME in this DI belongs. The parameter value must match the value of the LOG_GROUP parameter on the DEFINE_RECORDER_LOG_GROUP command in the System Configuration file for the DI that is the log recorder for this log group. The default log group name is CATENET (all DIs in the network). Each DI can belong to only one log group.

MESSAGE_NUMBER (MN)

List of message numbers that correspond to the set of messages to be logged by this DI. If this parameter is not specified, a CDCNET-defined set of log messages is selected for this DI to log. Refer to appendix D of the CDCNET Configuration and Site Administration Guide, Default Log and Alarm Messages, for the log message numbers and their message identifiers. You may add or delete log message numbers to this list using the CHANGE_SOURCE_LOG_GROUP command (see CDCNET Network Operations manual). However, omitting any of the default set of messages is not recommended. For the list of diagnostic messages and their numbers, refer to the CDCNET Diagnostic Messages manual.

Responses Source log group defined.

--ERROR-- A source log group is already defined for the system.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_source_log_group

define_source_log_group log_group=log_group_a
```

DEFINE_SYSTEM (DEFS)

Purpose Specifies the DI's logical name, defines values affecting the DI's memory management, and, for DIs supported by NOS hosts, specifies whether or not the DI contains the master clock for the network.

Format **DEFINE_SYSTEM**

SYSTEM_NAME = name
DATA_BUFFER_SIZE = 64..2304
BUFFER_PERCENTAGE = 1..99
BUFFER_BOUNDARY_PERCENTAGES = list 3 of integer 1..99
MEMORY_BOUNDARY_PERCENTAGES = list 3 of integer 1..99
MEMORY_MANAGER_PERIOD = 1..10
RESERVED_SYSTEM_SPACE = 1000..32768
STANDARD_STACK_SIZE = 0800(16)..2000(16)
DEFAULT_CHANNEL_TRUNK = name
ROUTING_SYSTEM = boolean
CLOCKING_SYSTEM = boolean

Parameters *SYSTEM_NAME (SN)*

Title of a DI, as it appears in the CDCNET directory. Default name is $DI_system_id, where system_id is the DI's system identifier consisting of 12 hexadecimal digits, as in 080025100068. An example of a default logical name is $DI_080025100068. If SYSTEM_NAME is specified, titles for both the specified system_name and the default system_name are registered for the DI. The system name appears on displays and is used in commands sent to the Network Operator Utility (NETOU).

DATA_BUFFER_SIZE (DBS)

Size, in bytes, of the system data buffers. Default value is 144 bytes. The value of this parameter is stored in battery-backed RAM and the effects are not realized until a reset other than a power-on reset occurs. Use extreme caution when changing this parameter value.

The actual buffer size generated is adjusted to be a multiple of a descriptor buffer. The following table defines the actual buffer sizes generated for ranges of entered data_buffer_size values.

DBS Value	Buffer Size	DBS Value	Buffer Size
64..70	68	1173..1210	1208
71..108	106	1211..1248	1246
109..146	144	1249..1286	1284
147..184	182	1287..1324	1322
185..222	220	1325..1362	1360
223..260	258	1363..1400	1398
261..298	296	1401..1438	1436
299..336	334	1439..1476	1474
337..374	372	1477..1514	1512
375..412	410	1515..1552	1550
413..450	448	1553..1590	1588
451..488	486	1591..1628	1626
489..526	524	1629..1666	1664
527..564	562	1667..1704	1702
565..602	600	1705..1742	1740
603..640	638	1743..1780	1778
641..678	676	1781..1818	1816
679..716	714	1819..1856	1854
717..754	752	1857..1894	1892
755..792	790	1895..1932	1930
793..830	828	1933..1970	1968
831..868	866	1971..2008	2006
869..906	904	2009..2046	2044
907..944	942	2047..2084	2082
945..982	980	2085..2122	2120
983..1020	1018	2123..2160	2158
1021..1058	1056	2161..2198	2196
1059..1096	1094	2199..2236	2234
1097..1134	1132	2237..2274	2272
1135..1172	1170	2275..2304	2310

BUFFER_PERCENTAGE (BP)

Sets the percentage of total System Main Memory (SMM) memory to be turned initially into data buffers. Default value is 50 percent.

BUFFER_BOUNDARY_PERCENTAGES (BBP)

Percentages of available buffers corresponding to boundaries between different levels of DI buffer availability. The DI dynamically maintains the state of available buffers. The four defined buffer states are GOOD, FAIR, POOR, and CONGESTED.

Specify a list of three integers that specify the three boundaries between the four buffer states. Default list value is (40, 20, 5). The first value defines the boundary value between GOOD and FAIR; the second value defines the boundary between FAIR and POOR; the third value defines the boundary between POOR and CONGESTED. Values must be listed from highest value to lowest. Values must differ by at least 5.

MEMORY_BOUNDARY_PERCENTAGES (MBP)

Percentages of available memory that correspond to boundaries between different levels of DI memory availability. The DI dynamically maintains the state of available memory. The four defined memory states are GOOD, FAIR, POOR, and CONGESTED.

Specify a list of three integers that specify the three boundaries between the four memory states. Default list value is (40, 15, 2). The first value defines the boundary value between GOOD and FAIR; the second value defines the boundary between FAIR and POOR; the third value defines the boundary between POOR and CONGESTED. Values must be listed from highest value to lowest. Values must differ by at least 5.

MEMORY_MANAGER_PERIOD (MMP)

Interval, in seconds, that the DI memory manager executes to maintain the DI buffer and memory state. Default is 1 second.

RESERVED_SYSTEM_SPACE (RSS)

Number of bytes to be reserved in the free memory pool for executive internal allocations. If specified as an odd value, this parameter is rounded off to the nearest integer divisible by two. Default is 1000 bytes.

STANDARD_STACK_SIZE (SSS)

Size, in bytes, of the task's stack size when the initiator of the task does not specify a stack size to the executive. If specified as an odd value, this parameter is rounded off to the nearest integer divisible by eight. Default is 2048 bytes.

DEFAULT_CHANNEL_TRUNK (DCT)

Specifies the default channel trunk for the configuration of NOS Network Product interface, gateways and Network Management Entities using NOS services. If a default channel trunk is not specified and the DI was loaded across an MCI interface, the trunk over which the DI was loaded becomes the default channel trunk. If a default channel trunk is not specified and the DI was not loaded across an MCI interface, the default channel trunk for the DI is not defined.

ROUTING_SYSTEM (RS)

Not used for this CDCNET release. The default value is FALSE.

CLOCKING_SYSTEM (CS)

Used only for DIs supported by NOS hosts. It indicates that the DI is to contain the master clock that specifies the date and time for the network. All other DI clocks set their date and time according to this master clock. The default value is FALSE. For DIs connected to a NOS host, there must be only one clocking system DI defined in the catenet with CLOCKING_SYSTEM=TRUE. For DIs supported by NOS/VE hosts, this parameter is not needed, since the DIs obtain the master clock from the NOS/VE host rather than from a clocking system DI. For an MDI/MTI connected to a NOS/VE host, the value for this parameter should be FALSE.

Responses The DEFINE_SYSTEM command can only be executed in a system configuration procedure. The following response listed is the only possible response if the DEFINE_SYSTEM command is entered through the Network Operator Utility (NETOU):

--ERROR-- The system is already defined.

The following responses may be logged during DI system startup:

The define system command is completed.

--WARNING-- System definition accepted with system not the clocking_ system.
The system could not be started as the master clock.

--WARNING-- System definition accepted with system not the clocking_ system.
There is already a master clock in catenet.
Network ID: xxxxxx, System Id: xxxxxxxxxxxx

--WARNING-- The define system command is completed.
Power on reset <P1> used, please correct.

--ERROR-- Buffer_boundary_percentage values not decreasing or do not differ by 5.
The buffer boundary percentages are <P1 P2 P3>.

--ERROR-- Memory_boundary_percentage values not decreasing or do not differ by 5.
The memory boundary percentages are <P1 P2 P3>.

--FATAL-- The system name cannot be registered.

Remarks This command is not required to be present in a DI's system configuration file. Default values are internally generated during initialization if this command is not present.

Proceed with caution if you use values other than the default values for any of the memory management parameters (DATA_BUFFER_SIZE through STANDARD_STACK_SIZE). Changing these values may improve system performance but can significantly degrade performance as well.

To change values of a DI's operating system while the DI is operational, use the CHANGE_SYSTEM command.

Examples

```
define_system system_name=mdi_86
```

DEFINE_TCP_INTERFACE (DEFTI)

Purpose Configures the TCP interface (DOD's Transmission Control Protocol). This command is required if this DI is to support TCP/IP protocols.

Format **DEFINE_TCP_INTERFACE**

ACCEPT_STRATEGY = keyword value
ACK_PERCENTAGE = 0..100
MAX_BUFFERS = 1..65535
MAX_SEGMENT_SIZE = 1..4096
MAX_CONNECTIONS = 0..512
QUIET_TIME = 0..10000
RETRANSMIT_STRATEGY = keyword value
RETRANSMIT_TIME = 0..65535
SECURITY_CHECKING = keyword value
TIME_TO_LIVE = 0..255
START = boolean

Parameters *ACCEPT_STRATEGY (AS)*

Specifies the TCP segment accept strategy to be used. A TCP segment is a packet of data that contains a TCP header which is delivered by IP to its destination. The following keyword values are allowed:

Keyword Value	Description
IN_ORDER (IO)	Segments are accepted only in the exact order they are expected. All other segments are discarded. Using this parameter may cause performance degradation and increase the number of retransmitted segments.
IN_WINDOW (IW)	Segments are accepted if they fall within the current TCP window. All other segments are discarded.

Default is IN_WINDOW.

ACK_PERCENTAGE (AP)

Specifies the percentage of the receive window that must be full before an acknowledgment is issued. The default is 50.

MAX_BUFFERS (MB)

Specifies the maximum number of data bytes that TCP will hold for a connection for both directions of travel. The default value is 2048 bytes.

MAX_SEGMENT_SIZE (MSS)

Specifies the maximum segment size, in bytes, to be negotiated for each new connection. The default value is 536 bytes.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous TCP connections. This includes active and passive connections. The default value is 200 connections.

QUIET_TIME (QT)

Specifies the number of seconds that TCP must wait, after a connection has closed, before a connection with the same source and destination socket addresses can be opened again. A TCP socket is an IP address and a TCP port ID. This socket is used by TCP to identify a TCP user process. If TCP receives a connection attempt with a source and destination socket address that are currently in a quiet time state, TCP will not respond or acknowledge connection establishment. The default value is 20 seconds.

RETRANSMIT_STRATEGY (RS)

Specifies the TCP segment retransmission strategy to be used. The following keyword values are allowed:

Keyword Value	Description
BATCH (B)	All unacknowledged segments are retransmitted when the retransmission timer expires.
FIRST_ONLY (FO)	Only the first segment of a sequence of unacknowledged segments will be retransmitted when the retransmission timer expires.
ADAPTIVE (A)	Each connection starts in FIRST_ONLY mode. If a subsequent retransmission sequence causes TCP to perform batch retransmission as a series of retransmissions, then TCP switches to BATCH mode. This case detects the instance where the peer TCP is using an IN_ORDER accept strategy.

Default is ADAPTIVE.

RETRANSMIT_TIME (RT)

Specifies the initial number of seconds that TCP should wait for an acknowledgment before retransmitting a data segment. This value changes for an active connection as the actual round-trip time is learned. The default value is 3 seconds.

SECURITY_CHECKING (SC)

Specifies the security checking to be performed on all segments. The following keyword values are allowed:

Keyword Value	Description
NONE (N)	The security option supplied in IP datagrams is ignored.
USER_SPECIFIED (US)	The security option specified by the upper-level protocol in the passive or active connect request establishes the security level of the connection.
LEVEL_U (LU)	All connections must be at security level UNCLASSIFIED.

Keyword Value	Description
LEVEL_C (LC)	All connections must be at security level CONFIDENTIAL.
LEVEL_E (LE)	All connections must be at security level EFTO.
LEVEL_M (LM)	All connections must be at security level MMMM.
LEVEL_P (LP)	All connections must be at security level PROG.
LEVEL_R (LR)	All connections must be at security level RESTRICTED.
LEVEL_S (LS)	All connections must be at security level SECRET.
LEVEL_T (LT)	All connections must be at security level TOP SECRET.

Default is NONE.

If a security level is specified, all connections and all segments received on a connection must match that security level. Any data segments that do not match the security level for a connection are discarded.

TIME_TO_LIVE (TTL)

Specifies the IP time-to-live field used by TCP. This is a hop count that is decremented at each move (hop) of a datagram. When the hop count reaches zero, the datagram is purged to prevent looping. The default value is 60 hops.

START (S)

Specifies that the TCP task should be started and connection attempts are honored. The default value is TRUE. For CDCNET 1.2.5, the only supported value is TRUE.

Responses

TCP Interface is defined and started.

--WARNING-- TCP Interface is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_tcp_interface

TCP Interface is defined and started.
```

DEFINE_TCPIP_GW (DEFTG)

Purpose Configures a gateway that provides services to NOS and NOS/VE CYBER-resident TCP/IP applications, such as FTP and SMTP. Gateways for the TCP, IP, and TELNET protocols can be configured. This command is not needed to support TELNET interactive services.

Format

DEFINE_TCPIP_GW
GATEWAY_NAME = name
SOURCE_IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = keyword value
PROTOCOL = keyword value
MAX_MESSAGE_SIZE = 1..65535 or keyword value
MAX_CONNECTIONS = 0..65535 or keyword value
START = boolean

Parameters

GATEWAY_NAME (GN)

The logical name of the gateway used in subsequent commands that reference the gateway.

SOURCE_IP_ADDRESS (IA)

The IP address of the CYBER host for which this gateway provides service. If this parameter is specified, host applications cannot specify their own source addresses in protocol connection requests. If omitted, the source address specified by the application is used. If the application also does not specify an address, then all requests are issued with an unspecified source address and IP uses the default source address for this DI. The format is similar to the decimal octet convention used by TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

For CDCNET Release 1.2.5, this parameter should not be specified.

TITLE (T)

Specifies the title which the host applications must use to access this gateway. This title must be coordinated with the configuration of the NDL OUTCALL and DDN Supervisor on NOS, or the Internet Protocol Access Method (IPAM) installation on NOS/VE. For CDCNET 1.2.5, it is strongly recommended that host applications specify the gateway title as GW_TCPIP_xxxx_yyyy, where xxxx is the mainframe model and yyyy is the serial number of each CYBER host for which this gateway provides services. The default value is the value supplied for the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the catenet for which gateway services are to be made available. The default is CATENET. For CDCNET 1.2.5, the only supported value is CATENET.

PROTOCOL (P)

Specifies the protocol supported by this gateway. The allowed keyword values are TCP, IP, and TELNET, which provide host interfaces for the respective protocols. This parameter controls which piece of the gateway is loaded. The default value is TCP. The IP and TELNET gateway interfaces are not supported for Release 1.2.5.

MAX_MESSAGE_SIZE (MMS)

Specifies the maximum number of bytes that a complete message can contain. If INFINITE is specified, there is no limit to the message size. The default value is 65535 bytes.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous connections to be supported by this gateway. If INFINITE is specified, there is no limit to the number of connections. The default value is 65535 connections.

START (S)

Specifies that this gateway is to be started. The default value is TRUE.

Responses

TCP/IP gateway <gateway_name> is defined and started to support <protocol> protocol.

TCP/IP gateway <gateway_name> is defined.

--ERROR-- TCP/IP gateway <gateway_name> is already defined.

--ERROR-- TCP/IP gateway title <title> already defined.

--FATAL-- TCP/IP gateway <gateway_name> was unable to open SAP.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_tcpip_gw gateway_name = ftp_gw title = gw_tcpip_0930_0106 ..
protocol = tcp

TCP/IP gateway FTP_GW is defined and started to support TCP protocol.
```

DEFINE_TIP (DEFT)

Purpose

Defines a single TIP in terms of (1) a CDCNET TIP name or an optional user TIP name that sites use to redefine TIP names, and (2) a set of default TIP parameters to be used if the identical parameters are not supplied on the command to configure the line and terminal devices. Identical parameters on the DEFINE_LINE and DEFINE_TERMINAL_DEVICE commands override the parameter settings on the DEFINE_TIP command.

You can specify certain parameters based on a TIP, line, or terminal device. These overlapping parameter definitions allow you to set values for these parameters on a TIP basis rather than specifying individual parameter definitions for each line that the TIP is to support.

To define the X.25 Asynchronous TIP in a DI, use the DEFINE_X25_ASYNCTIP command instead of DEFINE_TIP.

Format

DEFINE_TIP
TIP_NAME = keyword value
USER_TIP_NAME = name
LINE_CONTROL_SUPPORT = keyword value
FRAMING_TYPE = keyword value
CLUSTER_ADDRESS = 0..255
DEVICE_ADDRESS = 0..255
TRANSMISSION_BLOCK_SIZE = 128..4095
TERMINAL_USER_PROCEDURE = name

Parameters

TIP_NAME (TN)

Specifies the name of the tip defined by CDCNET. The following TIP names are allowed. Note that the suffix TIP may be omitted from a specified TIP name.

ASYNCTIP or ASYNC
HASPTIP or HASP
URITIP or URI
BSC3270TIP or BSC3270
BSCNJEFTIP or BSCNJEF
NTFTIP or NTF
USER1TIP or USER1
USER2TIP or USER2
USER3TIP or USER3
USER4TIP or USER4
XPCTIP or XPC
MODE4TIP or MODE4

For the XPCTIP, all other parameters on the DEFINE_TIP command are ignored.

If X.PC users must initially connect to the asynchronous ASCII TIP before switching to the XPCTIP, then the asynchronous TIP must also be defined. (Refer to the DEFINE_LINE command, parameter TIP_NAME, for more information.)

The XPCTIP is loaded when the first terminal user wants to use it or when a line configured with the XPCTIP becomes active. To reduce the wait-time for the first X.PC user, you can insert the following commands into the TDI's configuration procedure:

```
LOAD_MODULE XPCTIP_MODULE
```

This command forces the XPCTIP module to be loaded into the TDI during configuration, and to remain loaded after the last X.PC user disconnects.

```
LOAD_MODULE XPC_CP_MODULE
```

This command forces both the terminal user command processor module for X.PC and the XPCTIP module to be permanently loaded into the TDI.

USER_TIP_NAME (UTN)

This parameter is used for sites that implement user TIPs (TIPs developed at the site rather than provided with CDC release software) and wish to assign a site-defined logical name to a user TIP. The USER_TIP_NAME parameter is ignored if the TIP_NAME parameter does not signify one of the user TIPs. When the USER_TIP_NAME parameter is specified, all subsequent commands that have the TIP_NAME parameter (that is, all DEFINE_LINE commands) must also specify the value of the USER_TIP_NAME parameter.

LINE_CONTROL_SUPPORT (LCS)

Specifies the level of line control required by the TIP of LCM. The following keyword values are allowed.

Keyword Value	Description
NONE	Specifies that the control of the line is entirely the responsibility of the TIP.
CONFIGURATION	Specifies that the TIP expects the line control module (LCM) to perform the CIM configuration of the line, but that the TIP will monitor the line's modem signals after the line is configured.
FULL	Specifies that the TIP expects LCM both to configure the line and to monitor/process its modem signals.

The default value is FULL.

For CDC TIPs provided for this release of CDCNET, LCS must be FULL; otherwise lines supported by these TIPs are not enabled.

FRAMING_TYPE (FT)

Specifies the default framing to be used for this TIP. The following framing types are allowed:

ASYNC
SYNC
SDLC
PARALLEL

The following default framing types are set for each TIP type:

TIP Name	Default Framing Type
ASYNCTIP, XPCTIP	ASYNC
HASPTIP, BSCNJEFTIP, NTFTIP, BSC3270TIP, MODE4TIP	SYNC
URITIP	PARALLEL
Non-CDC provided TIPs	ASYNC

CLUSTER_ADDRESS (CA)

Specifies the default cluster address to be used by the TIP for communication with devices on lines supported by the TIP. This parameter is used by the HASP, BSC3270, and MODE4 TIPs only. For the HASP TIP, only one cluster is allowed on each line.

For the MODE4 TIP, the cluster address must be in the range of 70(16) through 7F(16) for Mode 4A clusters, and 20(16) through 7F(16) for Mode 4C clusters. The default cluster address for MODE4 TIP is 70(16).

Because the MODE4 TIP uses "cluster polling" for Mode 4C clusters, and since devices on a cluster are not auto-recognized, the configuration for each Mode 4C cluster should contain a DEFINE_TERMINAL_DEVICE command for each device in the cluster. Input from a device that is not configured results in a cluster failure.

The value on this command can be overridden by the CLUSTER_ADDRESS parameter on a DEFINE_TERMINAL_DEVICE (DEFTD) or DEFINE_BATCH_DEVICE (DEFBD) command. For this command, this parameter provides a mechanism for DEFTD/DEFBD parameters and/or commands to become optional. The default value is 0 unless a particular TIP sets it otherwise.

DEVICE_ADDRESS (DA)

Specifies the default device address to be used by the TIP for communication with devices on lines supported by the TIP. This parameter is used by the HASP, BSC3270, and MODE4 TIPs only. For this command, this parameter provides a mechanism for DEFINE_BATCH_DEVICE and DEFINE_TERMINAL_DEVICE parameters and/or commands to become optional. The value on this command can be overridden by the DEVICE_ADDRESS parameter on a DEFINE_TERMINAL_DEVICE (DEFTD) or DEFINE_BATCH_DEVICE (DEFBD) command.

For devices supported by the HASP TIP, the DEVICE_ADDRESS parameter is ignored if DEVICE_TYPE=CONSOLE, and need not be specified. Only one console is allowed per cluster or line. All other device types must have a device address ranging from 1 through 7, corresponding to the stream number of the HASP workstation device being configured. If not specified, the DA parameter will default to 1 for HASP batch devices.

For the MODE4 TIP, the DEVICE_ADDRESS parameter must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. The default device address for the MODE4 TIP is 61(16).

The default device address is TIP-dependent. The DA parameter normally defaults to 0 unless a particular TIP sets it otherwise.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the TIP for communication with devices on the lines supported by the TIP. The default value for this parameter varies according to the TIP being defined:

TIP Name	Default Transmission Block Size
HASPTIP, NTFTIP, BSC3270TIP	400
ASYNCTIP	450
BSCNJEFTIP	800
MODE4TIP	1040

This parameter is ignored if TIP_NAME = URITIP. The value of this parameter on this command can be overridden for a specific line or device by specifying the TBS parameter on the DEFINE_LINE, DEFINE_REMOTE_SYSTEM, DEFINE_BATCH_STREAM, DEFINE_BATCH_DEVICE, and/or DEFINE_TERMINAL_DEVICE command(s).

TERMINAL_USER_PROCEDURE (TUP)

Specifies the terminal user procedure (TUP) to be executed when a communication line supported by this TIP becomes active. A terminal user procedure file may contain most of the terminal user commands. This parameter provides the capability to predefine a user's terminal environment on a TIP basis and to have the environment automatically set up at the time a line supported by this TIP becomes active. There is no default for this parameter. The value of this parameter on this command may be overridden for a specific line or device by the TUP parameter on the DEFINE_LINE, DEFINE_REMOTE_SYSTEM, and/or DEFINE_TERMINAL_DEVICE command(s). This parameter is ignored if TIP_NAME = URITIP.

Responses

TIP <name> defined.

--ERROR-- TIP <name> already defined.

--ERROR-- Unable to define TIP <TIP name>. No CIM is installed.

--ERROR-- TIP <name> is not a CDC defined TIP name.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_tip tip_name=HASPTIP
```

DEFINE_USER_TELNET_GW (DEFUTG)

Purpose

Configures a user TELNET gateway, which provides CDCNET terminal users with access to the interactive services of remote hosts on a TCP/IP network. A site must configure a user TELNET gateway for each host that CDCNET terminal users access.

The two timeout parameters relate to the TCP and TELNET protocols respectively, and impose no limits on the user. That is, a user can leave a connection idle for any period of time without losing the connection. Note that the host service may impose inactivity limits of its own.

Format

DEFINE_USER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = keyword value
MAX_CONNECTIONS = 0..65535 or keyword value
SOURCE_IP_ADDRESS = list 4 of 0..255
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..7fffffff(16)
TCP_TIMEOUT = 0..65535 or keyword value
INACTIVITY_TIMEOUT = 0..65535
START = boolean

Parameters

GATEWAY_NAME (GN)

The logical name of the user TELNET gateway used in subsequent commands that reference the gateway.

IP_ADDRESS (IA)

The IP address of the host which provides the TELNET interactive service. This user TELNET gateway establishes a connection using this IP address as the destination address. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7).

TITLE or TITLES (T)

Specifies the title(s) by which this gateway service can be accessed. For example, this is the name that CDCNET terminal users supply in the CREATE_CONNECTION command. The default value is the value supplied for the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the CDCNET catenet that can access this service. For CDCNET Release 1.2.5, the only supported value is CATENET. The default value is CATENET.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous connections to be supported by the gateway. If INFINITE is entered, there is no restriction on the number of connections allowed. The default value is INFINITE.

SOURCE_IP_ADDRESS (SIA)

Specifies the IP address of the source host to be used by this gateway. The format is similar to the decimal octet convention used by the TCP/IP community, except the periods are replaced with commas and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 is represented as (128,2,53,7). The default value is the IP address from the host_type LOCAL DEFINE_IP_HOST command.

TCP_PORT_NUMBER (TPN)

Specifies the TCP port number to be used by the gateway. User TELNET issues a TCP active_connect request using the service contact port (the well-known port) for the destination port. The default is the well-known server TELNET port 23.

TCP_ALLOCATE_SIZE (TAS)

Specifies the amount of data that the gateway queues for each connection. Larger values may improve user response time, especially for PC users (with a standard protocol such as XMODEM), but can increase the number of instances of DI congestion. Specifying this value is discouraged, and should be done with caution, as poor network service may result. The default value is 4096 bytes.

TCP_TIMEOUT (TT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgment of data transmission. If an acknowledgment is not received within the specified period, TCP aborts the connection. A small value (less than a few seconds) may cause frequent and unnecessary loss of service during periods of network congestion. A large value may leave users waiting a long period of time after a host or network has failed. If INFINITE is entered, the connection does not timeout. The default value is 300 seconds.

INACTIVITY_TIMEOUT (IT)

Specifies the interval (in seconds) between inactivity checks. If a connection has been idle for the specified time, the gateway sends a TELNET status request to the remote TELNET to determine if the connection is still usable. The default is 600 seconds.

START (S)

Specifies that the newly configured gateway is to be started after it is defined. The default value is TRUE.

Responses

User TELNET gateway <gateway_name> is defined and started.

User TELNET gateway <gateway_name> is defined.

--ERROR-- User TELNET gateway <gateway_name> is already defined.

--ERROR-- User TELNET title <title> is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_user_telnet_gw gateway_name = gw_to_vax ..
ip_address = (128,5,0,3) title=vax
```

User TELNET gateway GW_TO_VAX is defined and started.

DEFINE_X25_ASYNCTIP (DEFXA)

Purpose

Defines X.25 Asynchronous TIP support for one or more X.25 trunks, and defines the set of default TIP parameters for those trunks. The services defined for asynchronous terminals connected through X.3 Packet Assembler/Disassemblers (PADs) are similar to those provided by the Asynchronous TIP to terminals connected through asynchronous lines.

This command includes terminal usage parameters that, for lines controlled by the Asynchronous TIP, are defined through the DEFINE_LINE command. DEFINE_X25_ASYNCTIP includes these parameters since the virtual circuits by which terminal users establish connections to the X.25 Asynchronous TIP are determined dynamically. This means that defining terminal usage on the basis of virtual circuit (equivalent to definition by DEFL) is not possible. Instead, this command defines these usage parameters for all virtual circuit connections for the trunks named in the command. Since the DEFXA command may be specified for each trunk, different usage parameters may be specified per trunk.

Format

DEFINE_X25_ASYNCTIP
  TRUNK_NAME = list 1..32 of name
  TERMINAL_DEFINITION_PROCEDURE = name
  TERMINAL_USER_PROCEDURE = name
  PROCEDURE_FILE_OPTION = keyword value
  CALLED_DTE_ADDRESS_RANGE = range of 1..15
  CONNECTION_CONNECT_TIMEOUT = 20..1000 or keyword value
  CONNECTION_DISCONNECT_TIMEOUT = 0..1000 or keyword value
  USER_CONNECTION_LIMIT = 1..16
  ACCEPT_REVERSE_CHARGES = boolean
  START = boolean

Parameters

TRUNK_NAME (TN)

Specifies the logical names of one or more X.25 trunks to be serviced by the X.25 Asynchronous TIP.

TERMINAL_DEFINITION_PROCEDURE (TDP)

Specifies the name of a terminal definition procedure (TDP) file. The commands within the named file will be executed when a virtual circuit becomes active. The default is that no TDP will be automatically executed for the virtual circuit.

TERMINAL_USER_PROCEDURE (TUP)

Specifies the name of a terminal user procedure (TUP) file. The commands within the named file will be executed each time a virtual circuit connection to a terminal becomes active. This parameter is ignored if a TDP (see TDP parameter) is specified. The default is that no TUP will be automatically executed for the virtual circuit.

PROCEDURE_FILE_OPTION (PFO)

This parameter allows for additional options for configuring terminal devices and attributes. The following keyword values are allowed:

Keyword Value	Description
LOGICAL_CHANNEL_ CONCATENATION (LCC)	Results in the X.25 logical channel number (LCN) being appended to the TDP's or TUP's file name, as in TDP_lcn or TUP_ lcn.
CALLED_DTE_ CONCATENATION (CDC)	Results in the called DTE address being appended to the TDP's or TUP's file name, as in TDP_called_dte_address.
CALL_DATA_PROCEDURE (CDP)	Selects the option to treat call data information as if it were a TDP file name.

The default is that the call request will not specify any procedure name information.

CALLED_DTE_ADDRESS_RANGE (CDAR)

Specifies the range of the called dte address to be used for concatenation when the CALLED_DTE_CONCATENATION option is selected for the PROCEDURE_FILE_OPTION parameter. This parameter is ignored if PROCEDURE_FILE_OPTION is not equal to CDC. The default is that the entire DTE address will be used.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines how much time the terminal user has to create the first $input/$output connection. If no connection is established within that time, the virtual circuit will be cleared. The range is 20 through 1000 seconds. Default is 120 seconds. The keyword INFINITE indicates an infinite time. The timer has a precision of +/- 2 seconds.

CONNECTION_DISCONNECT_TIMEOUT (CDT)

Defines how much time the terminal user has to establish a new $INPUT/$OUTPUT connection after the last such connection is disconnected. If no new connection is established within that time, the virtual circuit will be cleared. The range is 0 through 1000 seconds, default is 120. The keyword INFINITE indicates an infinite time. The timer has a precision of +/- 2 seconds.

USER_CONNECTION_LIMIT (UCL)

Defines the maximum number of $INPUT/$OUTPUT connections allowed at any one time by a user. The range is 1 through 16 connections. Default is 4.

ACCEPT_REVERSE_CHARGES (ARC)

Specifies whether or not the X.25 Asynchronous TIP should accept incoming calls with reverse charges specified. The default value is false, that is, reverse charges are not accepted.

START (S)

Specifies whether or not the configured X.25 Asynchronous TIP should begin to accept incoming calls for terminal connections. The default value is true; started.

Responses

X.25 AsyncTip support defined for specified trunks.

X.25 AsyncTip support defined and started for specified trunks.

--ERROR-- Trunk <trunk_name> is not a X.25 trunk.

--ERROR-- X.25 Interface has not been defined (DEFXI) on trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> already assigned X.25 AsyncTip support.

--ERROR-- Duplicate trunk name <trunk_name> specified.

--FATAL-- Not enough memory available for required table space.

Examples

```
define_x25_asynctip trunk_name=x25_telenet
```

DEFINE_X25_GW (DEFXG)

Purpose

This command and the ADD_X25_GW_OUTCALL (ADDXGO) command define X.25 application gateway service for one or more X.25 trunks. This command establishes a logical association between the trunks, enabling them to be viewed as one logical link to the public data network (PDN).

The command defines the X.25 protocol IDs for which incoming calls are accepted by the gateway for these trunks. Two parameters specify the protocol IDs: One for calls to be routed to NOS systems; and one for calls to be routed to NOS/VE systems or other gateways. This command, and the ADDXGO command, also define the CDCNET titles by which this gateway definition (the association of trunks) is known throughout CDCNET. The ADDXGO command defines titles by which NOS/VE applications or other gateways may direct outgoing calls to the X.25 trunks defined for the gateway. The X.25 addressing is supplied for the application on the ADDXGO command when using these titles.

Additional titles are specified by two parameters on the DEFXG command. One parameter specifies a list of port numbers by which NOS applications may direct outgoing calls to the associated trunks. NOS applications making calls through this gateway must construct their OUTCALL blocks with a PORT field equal to one of the defined port numbers. The other parameter specifies a list of site-defined titles by which NOS/VE applications may direct outgoing calls to the X.25 trunks defined for the gateway.

Refer also to the ADD_X25_GW_OUTCALL command description in this chapter.

Format

DEFINE_X25_GW
GATEWAY_NAME = name
TRUNK_NAME = list 1..32 of name
NOS_PROTOCOL_ID = list 1..7 of 2..255
CDCNET_PROTOCOL_ID = list 1..7 of 2..255
NOS_PORT_NUMBER = list 1..15 of 1..255
VE_OUTCALL_TITLE = list 1..15 of name
START = boolean
DTE_ADDRESS_PROTOCOL_ID = list 1..7 of 2..255

Parameters

GATEWAY_NAME (GN)

Specifies the logical name of the gateway to be used in subsequent commands that reference the gateway. This gateway name must be unique within the catenet.

TRUNK_NAME (TN)

Specifies the logical name or names of one or more X.25 trunks to be serviced by the X.25 gateway. The X.25 trunks must all belong to the same PDN. If trunks for other PDNs are to be serviced by an X.25 gateway, separate gateway definitions for each PDN must be specified.

NOS_PROTOCOL_ID (NPI)

Specifies one or more protocol IDs that identify incoming calls to be routed to NOS mainframes by host node number. The default NOS protocol IDs are C0(16) and C1(16).

CDCNET_PROTOCOL_ID (CPI)

Specifies one or more protocol IDs that identify incoming calls to be routed to NOS/VE systems or other gateways by application title. The default CDCNET protocol ID is C2(16).

NOS_PORT_NUMBER (NPN)

Specifies a list of NOS port numbers used for NOS outgoing calls to the X.25 network. The NOS applications access the X.25 network supported by this gateway definition by constructing OUTCALL blocks with a PORT field value equal to a defined port number. If no NOS port numbers are specified, NOS mainframes are not able to access this gateway.

VE_OUTCALL_TITLE (VOT)

Specifies a list of titles that NOS/VE applications can use to access this gateway. These titles are used to support outgoing calls by NOS/VE applications to the X.25 network supported by this gateway definition. If this parameter is not specified, no VE outcall titles are registered. This parameter is not used by NOS-only environments.

START (S)

Specifies whether or not the gateway should be started after it is configured. The default value is TRUE; started.

DTE_ADDRESS_PROTOCOL_ID (DAPI)

Specifies one or more protocol IDs which identify incoming calls to be routed to CDNA server applications by constructing a title from the protocol ID and the called DTE address. The default is that the protocol ID and the DTE address, by themselves, cannot identify the called application.

Responses

X.25 gateway <gateway_name> is defined.

X.25 gateway <gateway_name> is defined and started.

--ERROR-- X.25 gateway <gateway_name> is already defined.

--ERROR-- The specified X.25 trunks do not connect to the same PDN.

--ERROR-- X.25 gateway <gateway_name> is not defined.

--ERROR-- X.25 gateway <gateway_name> Protocol ID already assigned.

--ERROR-- Trunk name <trunk_name> may not appear more than once in the list of specified trunks.

--ERROR-- Protocol ID <protocol_id> may not appear more than once in the list of protocol IDs.

--ERROR-- VE_outcall_title <name> may not appear more than once in the list of specified VE Outcall Titles.

--ERROR-- NOS_port_number <value> may not appear more than once in the list of specified NOS Port Numbers.

--FATAL-- X.25 gateway - not enough memory available for required table space.

Examples

```
define_x25_gw trunk_name=X25TELENET,nos_port_number=05(16)
```

DEFINE_X25_INTERFACE (DEFXI)

Purpose

Defines an X.25 interface (Packet Level Interface) which can then be used to support an X.25 network solution or X.25 gateway. This command includes parameters that define the ranges for permanent, incoming-only, two-way, and outgoing-only virtual circuits. Although the PVC_RANGE, INONLY_RANGE, TWOWAY_RANGE, and OUTONLY_RANGE virtual circuit range parameters are all optional, at least one range must be specified if the command is to execute successfully. If more than one range is specified, the associated PVC_RANGE, INONLY_RANGE, TWOWAY_RANGE, and OUTONLY_RANGE value ranges must be in ascending order, with no overlapping of the value ranges.

Format

DEFINE_X25_INTERFACE
TRUNK_NAME = name
PUBLIC_DATA_NETWORK = name or keyword value
INTERFACE_NAME = name
LOCAL_DTE_ADDRESS = 1..15 of string
PACKET_SEQUENCE_NUMBERING = keyword value
PVC_RANGE = range of 1..4095
INONLY_RANGE = range of 1..4095
TWOWAY_RANGE = range of 1..4095
OUTONLY_RANGE = range of 1..4095
DEFAULT_WINDOW_SIZE = 1..127
DEFAULT_PACKET_SIZE = keyword value
CONGESTED_THRESHOLD = 25..255
START = boolean

Parameters

TRUNK_NAME (TN)

Specifies the name of the trunk to be used by the X.25 interface. This name must be unique in the catenet and must be coordinated with the trunk name specified on the DEFINE_X25_TRUNK command.

PUBLIC_DATA_NETWORK (PDN)

Specifies the name of the PDN for the interface. Allowed keyword values include the following:

TELENET
TYMNET
UNINET
CDSN
DATAPAC
TRANSPAC
USERPSN1
USERPSN2
USERPSN3
USERPSN4

INTERFACE_NAME (IN)

Logical name of the X.25 interface. This name is used to refer to the X.25 interface in subsequent commands. The default INTERFACE_NAME is the TRUNK_NAME value.

LOCAL_DTE_ADDRESS (LDA)

Specifies the local data terminal equipment (DTE) address associated with this X.25 interface. This parameter is specified as a string of digits with values ranging from 0 through 9, and this string must match the DTE address assigned at subscription time. The local DTE address may be used by applications in outgoing call requests to specify the X.25 interface and trunk over which the call should be made. If this parameter is not specified, the X.25 interface cannot be selected by applications using the local DTE address.

PACKET_SEQUENCE_NUMBERING (PSN)

Specifies the X.25 packet sequence numbering to be used for this interface. The following keyword values are allowed.

Keyword Value	Description
NORMAL	Specifies normal X.25 packet numbering performed modulo 8.
EXTENDED	Specifies extended packet numbering performed modulo 128.

Default is NORMAL.

The PSN must be coordinated with the optional user facilities selected at subscription time.

PVC_RANGE (PR)

Specifies the range of logical channel numbers to be used for permanent virtual circuits. This parameter must match subscription time value.

INONLY_RANGE (IR)

Specifies the range of logical channel numbers to be used for incoming calls only. This parameter value must match subscription time value.

TWOWAY_RANGE (TR)

Specifies the range of logical channel numbers to be used for either incoming or outgoing calls. This parameter value must match subscription time value.

OUTONLY_RANGE (OR)

Specifies the range of logical channel numbers to be used for outgoing calls only. This parameter value must match subscription time value.

DEFAULT_WINDOW_SIZE (DWS)

Specifies the window size to be used for virtual calls for this interface if window size is not negotiated in the virtual calls. Default DWS value is 2. This parameter value must match subscription time value.

DEFAULT_PACKET_SIZE (DPS)

Specifies the data packet size to be used for virtual calls for this interface if packet size is not negotiated in the virtual calls. The following packet sizes are allowed.

16
32
64
128
256
512
1024

Default DPS value is 128. The DPS must match subscription time value.

CONGESTED_THRESHOLD (CT)

Specifies the number of messages in the interface outgoing queue at which the interface is considered congested. The default value is 40. For CDCNET 1.2 and future releases, this parameter will be ignored.

START (S)

Specifies whether or not the configured element should be started. The default value is TRUE; started.

Responses

X.25 Interface <interface_name> defined and started.

X.25 Interface <interface_name> defined.

--ERROR-- X.25 Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not an X.25 Trunk.

--ERROR-- X.25 Interface already defined for trunk <trunk_name>.

--ERROR-- X.25 Interface <interface_name> is already defined.

--ERROR-- Expecting digit in DTEA found = <value>.

--ERROR-- Default packet size of <value> is not a valid X.25 packet size.

--ERROR-- No logical channel assignments defined.

--ERROR-- Specified channel assignments result in overlapping pvc, in-only, two-way, or out-only channels.

--ERROR-- Default window size may not be greater than 7 for NORMAL packet sequence numbering.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_x25_interface trunk_name=X25TELENET ..
public_data_network=telenet interface_name=X25TEL twoway_range=1..32
```

DEFINE_X25_NET (DEFXN)

Purpose

Configures a CDCNET X.25 network solution, using a virtual circuit on a previously defined X.25 trunk.

Format

DEFINE_X25_NET
TRUNK_NAME = name
REMOTE_DTE_ADDRESS = 1..15 of string
NETWORK_ID = 1..7FFFFFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
ROUTING_INFO_NETWORK = boolean
NETWORK_PROTOCOL_ID = 2..255
ACCEPT_PDN_CHARGES = boolean
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..500000

Parameters

TRUNK_NAME (TN)

Specifies the name of the X.25 trunk to use for the network solution. The trunk name must be the one defined for the trunk by the TRUNK_NAME parameter on the DEFINE_X25_TRUNK command.

REMOTE_DTE_ADDRESS (RDA)

Specifies the remote data terminal equipment (DTE) address for this X.25 network. This parameter is specified as a string of digits with values of 0 through 9. The RDA is called when the X.25 network is established from this system. The calling address on an X.25 call indication received for this network is also validated against the RDA address. A call with an invalid calling address is cleared.

NETWORK_ID (NI)

CDCNET network identifier of the X.25 network solution. The network ID must be unique within the catenet.

NETWORK_NAME (NN)

Logical name of the network solution used in subsequent commands referencing the network solution. The default name is constructed from the network ID parameter, using the format *$NET_xxxxxxxx*, where *xxxxxxxx* is the network_id expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

Cost of the network solution. The cost of a network may be calculated by dividing 1,000,000 by the data rate of the network in bits per second. Cost is used by CDCNET network routing to determine the least-cost routes to use to interconnect networks. For example, the cost of a trunk with a speed of 56,000 bits per second is 06FA(16).

RELAY_ALLOWED (RA)

Indicates whether relay is allowed through this network solution. If RA is TRUE, this network may be used as part of a route to interconnect two other networks. If RA is FALSE, this network may be used only as part of an interconnecting route when no other route can be used to interconnect the networks. The default for an X.25 network is FALSE; relay not allowed.

ROUTING_INFO_NETWORK (RIN)

Indicates whether or not the network solution is to carry CDCNET routing information. If RIN is true, routing information describing all the networks to which this system is attached is sent over this network solution. If RIN is false, routing information is not sent by this system over the network solution. This system would appear, then, as not connected to any network other than this network solution. The default value is TRUE; network solution carries CDCNET routing information.

NETWORK_PROTOCOL_ID (NPI)

Specifies the protocol ID that identifies incoming calls for the X.25 network. The default NPI ID is C3(16). If more than one DEFINE_X25_NET command is encountered for the same trunk name, each NPI value on each DEFINE_X25_NET command must be unique.

ACCEPT_PDN_CHARGES (APC)

If the value of APC is TRUE, this system may call the remote DTE address with normal charging and may accept a call with normal or reverse charging from the remote DTE address. If the value of APC is FALSE, this system may call only the remote DTE address with reverse charging requested and may accept only calls with normal charging from the remote DTE address. The default value is TRUE; accept PDN charges.

START (S)

Specifies whether or not the X.25 network solution should be started after it is configured. The default value is TRUE; started.

ARCHITECTURE_TYPE (AT)

Allowed architecture types are CDNA and DOD. The DOD parameter value is currently not supported. The default value is CDNA.

OUTPUT_QUEUE_LIMIT (OQL)

Specifies, in bytes, the maximum amount of data which is retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. The newest output messages are discarded first if messages need to be discarded.

The default value depends on the cost of the network (see COST parameter). If the cost is 6FA(16) or greater, then the default output queue limit is 30000 bytes. Otherwise, the default value is 60000 bytes.

Responses

X25 network <network_name> defined for trunk <trunk_name>.

X25 network <network_name> defined and started for trunk <trunk_name>.

--ERROR-- Network <network_name> already defined for trunk trunk_name>.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not a X.25 trunk.

--ERROR-- Network name <network_name> already defined.

--ERROR-- Network ID <network_id> already defined.

--ERROR-- Protocol ID <id> already assigned.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_x25_net trunk_name=X25TELENET,..
remote_dte_address='6124821234',network_name=X25NET,network_id=0a(16)
```

DEFINE_X25_TRUNK (DEFXT)

Purpose Configures an X.25 trunk by defining the layer 2 parameters of an interface to an X.25 network.

Format **DEFINE_X25_TRUNK**
LIM = 0..7
PORT = 0..3
TRUNK_NAME = name
MODE = keyword value
MAX_UNACK_FRAMES = 0..7
PF_RECOVERY_TIMER = 500..65535
RETRANSMISSION_LIMIT = 1..65535
TRUNK_SPEED = keyword value
CLOCKING = keyword value
INTERACTIVE_BANDWIDTH (IB) = 1..9

Parameters **LIM (L)**

LIM number for the port to which the X.25 line is connected.

PORT (P)

Port number for the port to which the X.25 line is connected.

TRUNK_NAME (TN)

Logical name of the X.25 trunk. The default name is constructed from the LIM and PORT parameters, as in $LIM3_PORT1.

MODE (M)

Mode of operation of the X.25 trunk. The following keyword values are allowed.

Keyword Value	Description
DCE	Specifies that the DI will operate as the Data Communications Equipment end for the X.25 trunk.
DTE	Specifies that the DI will operate as the Data Terminating Equipment end for the trunk.

Default mode is DTE.

MAX_UNACK_FRAMES (MUF)

Window size specifying the maximum number of frames the local station can send without receiving an acknowledgement (X.25 CCITT parameter K). Default value is 7 frames.

PF_RECOVERY_TIMER (PRT)

Value of the P/F recovery timer in milliseconds. This timer initiates recovery when an acknowledgement is not received within this time period (X.25 CCITT timer T1). The default value is 500 milliseconds.

RETRANSMISSION_LIMIT (RL)

Maximum retransmissions allowed. The default value is 5 retransmissions.

TRUNK_SPEED (TS)

Speed of the X.25 trunk in bits per second. Trunk speed is used by the LIM to generate the data clocking for the trunk (except when clocking has been specified to be EXTERNAL), and to configure the media with the proper values for the network cost and output queue limit. The possible values for this parameter are:

1200
2400
4800
9600
19200
38400
48000
56000
64000

Default is 56000. Failure to specify this parameter for any speed other than 56000 bits per second will result in suboptimal performance.

CLOCKING (C)

Specifies whether the LIM internally generates the clock signal for data on this trunk or uses an externally-generated clock signal for data on the trunk. If the LIM generates the data clock signal, the clocking rate is derived from the TRUNK_SPEED parameter. The following keyword values are allowed.

Keyword Value	Description
EXTERNAL	Specifies that the LIM derives data clocking for both receive and transmit data from external signals (TRUNK_SPEED is then informational only). The EXTERNAL receive data clock is derived from the RS232 DD circuit for RS232 ports, or the RS449 SR circuit for RS449 ports. The EXTERNAL transmit data is derived from the RS232 DB circuit for RS232 ports, or the RS449 ST circuit for RS449 ports.

Keyword Value	Description
TRANSMIT	Specifies that the LIM generates the clocking for transmit data, but derives the clocking for receive data from an external source. The transmit data clock matches the trunk speed specified for the line. The LIM supplies the transmit data clock on the RS232 DA circuit (for RS232 ports) or the RS449 TT circuit (for RS449 ports). The LIM derives the receive data clock from the RS232 DD circuit (for RS232 ports) or the RS449 SR circuit (for RS449 ports).

Default clocking is EXTERNAL.

Clocking should be TRANSMIT for X.25 trunks connected directly to terminal equipment (without intervening modems). Clocking should be EXTERNAL for X.25 trunks with modems.

INTERACTIVE_BANDWIDTH (IB)

Specifies the percentage of the trunk bandwidth to be used to transmit data at interactive priority. The default value is 7. For example, a value of 7 on this parameter will, on average, result in 70 bytes of interactive priority data for every 30 bytes of batch priority data.

Responses

X25 trunk <trunk_name> defined.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- The Device Interface does not contain a CIM board.

--ERROR-- Specified LIM, PORT is already in use.

--ERROR-- Specified LIM, PORT is not on.

--ERROR-- LIM <xx>, PORT <yy> is not installed in this system.

--ERROR-- LIM <xx>, PORT <xx> addresses a port that cannot be serviced. More than 48 ports are attached to CIMxx. Ports beyond the 48th port attached to a CIM are not serviced.

--ERROR-- Not enough CIM memory available to load <xxx> I/O Processor.

--ERROR-- Specified port value is greater than 1 for a 2-port LIM.

--ERROR-- Line_speed <integer> is not supported for an X.25 trunk.

--ERROR-- X.25 is not supported on the specified LIM.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_x25_trunk lim=0 port=0 trunk_name=X25TELENET
```

DI Configuration Procedure Commands - NOS/VE Only

This section contains quick–reference command descriptions of commands that are used only for DI configuration procedures in a NOS/VE environment.

DEFINE_CHANNEL_NET (DEFCN)

Purpose Configures a CDCNET channel network solution to a NOS/VE system using a previously defined channel trunk. An "unable to start" error leaves the network defined but not started.

Format **DEFINE_CHANNEL_NET**
TRUNK_NAME = name
NETWORK_ID = 1..7FFFFFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
MULTICAST_NETWORK = boolean
ROUTING_INFO_NETWORK = boolean
CONGESTED_THRESHOLD = 20..255
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..50000

Parameters **TRUNK_NAME (TN)**

The logical name of the channel trunk to be used for the network solution. The channel trunk with this name must be configured prior to the execution of this command.

NETWORK_ID (NI)

The CDCNET network identifier of the channel network solution. This network identifier must match the network identifier defined in the corresponding NOS/VE configuration entry using the Logical Configuration Utility (LCU). The LCU subcommands that define channel network identifiers are DEFINE_CHANNEL_NETWORK (for non-CYBER 930 hosts) and DEFINE_NETWORK_ACCESS (for CYBER 930 hosts). Both subcommands assign the network identifier using the NETWORK parameter. For more information, refer to the Logical Configuration Utility chapter of the NOS/VE System Analyst Reference Set, System Performance and Maintenance.

NETWORK_NAME (NN)

The logical name of the network solution used in subsequent commands referencing the network solution. The default name is constructed from the NETWORK_ID parameter, using the format *$NET_xxxxxxxx*, where *xxxxxxxx* is the network ID expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

The cost of the network solution. The cost of a network may be calculated by dividing 100 million by the data rate of the network in bits per second. Cost is used by CDCNET network routing to determine the least cost routes to use to interconnect networks. The default cost of a channel trunk is 10(16), the cost of a 6-megabit trunk.

RELAY_ALLOWED (RA)

Indicates whether relay is allowed through this network solution. If RA is TRUE, than this network may be used as part of a route to interconnect two other networks. If RA is FALSE, then this network will be used only as part of an interconnecting route when no other route can be used to interconnect the networks. The default for a channel network is FALSE; relay not allowed.

MULTICAST_NETWORK (MN)

Indicates whether or not the network solution is a multicast network. The default value is TRUE; network solution is a multicast network.

ROUTING_INFO_NETWORK (RIN)

Indicates whether or not the network solution is to carry CDCNET routing information. If RIN is true, routing information describing all the networks to which this system is attached is sent over this network solution. If RIN is false, routing information is not sent by this system over the network solution. This system would appear, then, as not connected to any network other than this network solution. The default value is TRUE; network solution carries CDCNET routing information.

CONGESTED_THRESHOLD (CT)

Specifies the number of messages in the network solution outgoing queue at which the network solution is considered congested. The default value is 30. Note that the point at which the network is considered uncongested after being congested, is 75 percent of the CONGESTED_THRESHOLD. For this release and future releases, this parameter will be ignored.

START (S)

Specifies whether or not the configured element should be started. The default value is TRUE; started.

ARCHITECTURE_TYPE (AT)

Specifies the network architecture that this network supports. Allowed architecture types are CDNA and DOD. The DOD parameter value is currently not supported. The default value is CDNA.

OUTPUT_QUEUE_LIMIT (OQL)

Specifies, in bytes, the maximum amount of data which is retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. The newest output messages are discarded first if messages need to be discarded.

The default value depends on the cost of the network (see COST parameter). If the cost is 6FA(16) or greater, then the default output queue limit is 30000 bytes. Otherwise, the default value is 60000 bytes.

Responses

CHANNEL network <network_name> defined for trunk <trunk_name>.

CHANNEL network <network_name> defined and started for trunk <trunk_name>.

--WARNING-- The value specified for the network_id, <value>, is greater than 65535 (0ffff(16)). Future CDCNET releases will not support a Network_id greater than 65535 (0ffff(16)).

--WARNING-- The 3A Command Processor timed out waiting for a response from the SSR.
Please check network status for completion of requests.

--ERROR-- Network <network_name> already defined for trunk <trunk_name>.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not a CHANNEL trunk.

--ERROR-- Trunk <trunk_name> already assigned.

--ERROR-- Network name <network_name> already defined.

--ERROR-- Network id <network_id> already defined.

--ERROR-- Trunk <trunk_name> down.
Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> off.
Unable to start network <network_name>.

--FATAL-- Not enough memory is currently available for required table space.

--FATAL-- Stream Service error.
Not enough memory is currently available for required table space.

--FATAL-- Stream Service error.
Unable to open statistics SAP.

--FATAL-- Stream Service error.
Unable to open memory management SAP.

--FATAL-- Stream Service error.
Unable to initialize MCI board.

--FATAL-- Unable to start task <entry_point_name>.

Remarks

This command is not required for CDCNET 1.2.5. NOS/VE MDI/MTI systems loaded via a channel trunk to a NOS/VE CYBER system provide the DEFCN definition through the load process.

Examples

```
define_channel_net trunk_name=channel_trunk_2 ..
network_id=002(16)
```

DI Configuration Procedure Commands - NOS Only

This section contains quick–reference command descriptions of commands that are used only for DI configuration procedures in a NOS environment.

ADD_NP_GW_OUTCALL (ADDNGO)

Purpose

Creates a Network Products gateway outcall definition. Outcall is from the perspective of the CDCNET network; that is, the call is going out of the CDCNET network. Outcall information is used to generate the proper call request into the foreign network. A Network Products (NP) gateway outcall definition consists of an NP application name and the corresponding first type of CDCNET title by which the gateway is to be known throughout the CDCNET network. These titles are completely site-definable and conform to the CDCNET conventions for titles. Two examples of such titles are PTFS$M80 and QTFS$LID.

Refer also to the DEFINE_NP_GW command description in this chapter.

Format

ADD_NP_GW_OUTCALL
TITLE = any
NP_APPLICATION_NAME = string 1..7
GATEWAY_NAME = name

Parameters

TITLE (T)

The title that CDNA applications can use to access a particular NOS application through this gateway. The title is used to support calls from CDNA systems to the NOS host. The title must be unique to the gateway (including titles specified by the DEFINE_NP_GW command). The title can be of type name (up to 31 characters long, without quotation marks), or type string (up to 255 characters long, within quotation marks).

NP_APPLICATION_NAME (NAN)

The NOS application name that is accessed when a CDNA application (or gateway) initiates a call connection with the corresponding CDNA title.

GATEWAY_NAME (GN)

The name of the NOS gateway which provides access to the NOS application. The gateway must be previously defined. If this command is specified in a configuration file, the default value for this parameter is the previously defined NOS gateway name. If this command is entered by the network operator through the Network Operator Utility (NETOU), this parameter is required.

Responses

NP gateway title <title> added.

--ERROR-- Title <title> already defined.

--ERROR-- NP gateway <name> not defined.

--ERROR-- No NP gateway defined.

--ERROR-- <title> must be string or name type.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
add_np_gw_outcall title='PTFS$M80' ..
np_application_name='PTFS'

NP gateway title PTFS$M80 added.
```

DEFINE_FILE_SUPPORT (DEFFS)

Purpose

Defines the file access function to be present in this DI and selects the file types to be supported.

This command is optional in the system configuration procedures of MDIs loaded via MCI that provide file support for the network, using the Independent File Access ME. The host must be configured to run the Network File Server application, NETFS.

If this command is not present in an MDI's configuration file, then access to all CDCNET file types is supported. If file support is desired for a host whose MCI link was not used to load the MDI (that is, if multiple MCI boards are present in the MDI or MTI), then this command is required.

File support for each unique trunk name operates independently. That is, the file support for multiple trunk names operates as if the trunk names were defined on separate MDI/MTI systems. Commands addressed to the Independent File Access ME function for one trunk will not affect the Independent File Access ME function for other trunks.

Format

DEFINE_FILE_SUPPORT
FILE_TYPE = list 1..8 of keyword value
TRUNK_NAME = name

Parameters

FILE_TYPE or *FILE_TYPES (FT)*

Types of files to be supported at this MDI. Files can be specified as a list of one or more of the following keyword values:

Keyword Value	Description
EXCEPTION	Exception list for the network that contains loading and dumping specifications for downloaded DIs
BOOT	Boot files for DIs
DUMP	DI dump files
LIBRARY	CDCNET DI object library files
CONFIGURATION	DI configuration files and procedures
TERMINAL_PROCEDURE	Terminal definition procedures
USER_PROCEDURE	Terminal user procedures
LOAD_PROCEDURE	Vertical Format Unit load procedures
ALL	All file types

Default is ALL.

TRUNK_NAME (TN)

The trunk name of the logical link which is to be used for the file access connection. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified via a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk. If the MDI/MTI was not loaded over a channel and no default channel trunk was specified on a the DEFINE_SYSTEM command, then no default channel trunk exists.

Responses

File Support is defined for trunk <name>.

--WARNING-- File Support is defined for trunk <name>. NP interface for the trunk is started but the logical link is down.

--WARNING-- File Support is defined for trunk <name>. NP interface for the trunk is not started. Start NP interface to enable file support.

--ERROR-- NP interface is not defined for trunk <name>.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--ERROR-- File Support for trunk <name> is already defined.

--FATAL-- File Support cannot be defined for trunk <name>. Not enough memory is currently available for required table space.

--FATAL-- File Support cannot be defined for trunk <name>. Unable to initialize the File Support function.

Remarks

This command can be used to redefine the default file type support of an MDI or MTI loaded across an MCI. The first time file support is redefined for the DI load media, you will not receive the following error message:

--ERROR-- File Support for trunk <name> is already defined.

If this command is present in the MDI's configuration file, it will redefine the default file type support of an MDI loaded across an MCI. However, this command does not redefine file support in an MDI if it is entered through the Network Operator Utility (NETOU). To redefine file support via NETOU, you must first cancel file support for all file types using the CANCEL_FILE_SUPPORT command before redefining the desired file types using DEFINE_FILE_SUPPORT.

Examples

```
define_file_support file_types=(exception,boot,library)
```

DEFINE_NP_GW (DEFNG)

Purpose

This command and the ADD_NP_GW_OUTCALL (ADDNGO) command define the CDCNET titles by which this NOS Network Products application-to-application (A-to-A) gateway is to be known throughout CDCNET.

Three types of titles are associated with this gateway. The first type of title (defined by the ADDNGO command) supports access to a specific application on NOS via the NP gateway from NOS/VE applications or from another NP gateway.

The second type of title supports access to the gateway by using the coupler node number. This title is used by the X.25 gateway and optionally by the NP gateway. The format of this second title type is *$GW_NP_xx*, where *xx* is the ASCII representation of a two-digit hexadecimal number, which is specified by the NOS_DHOST_NUMBER parameter. Only the *xx* portion of the title is site-definable, and the *$GW_NP_* portion is internally supplied by CDCNET. Calling NOS applications must have their respective OUTCALL blocks constructed with a DHOST field set to the *xx* value registered as part of the title of the called NOS gateway.

The third type of title supports access in the same way as the second type, except the title is completely site-definable and conforms to the CDNA definition of a title. This type is currently not used.

Refer also to the ADD_NP_GW_OUTCALL command description in this chapter.

Format

DEFINE_NP_GW
GATEWAY_NAME = name
TRUNK_NAME = name
NOS_PROTOCOL_ID = list 1..7 of 2..255
CDCNET_PROTOCOL_ID = list 1..7 of 2..255
NOS_DHOST_NUMBER = list 1..15 of 0..0FF(16)
TITLE = list 1..15 of name
DEFAULT_TRANSLATION_DOMAIN = name
DEFAULT_SEARCH_DOMAIN = name
START = boolean

Parameters

GATEWAY_NAME (GN)

Logical name of the gateway used in subsequent commands that reference the gateway. If GATEWAY_NAME is not specified, the TRUNK_NAME parameter value will be assigned as the default gateway name.

TRUNK_NAME (TN)

Specifies the trunk name of the NOS host/MDI logical link which is to be used to support A-to-A traffic with this host. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified via a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk.

NOS_PROTOCOL_ID (NPI)

Specifies one or more protocol IDs that identify outcalls to be routed to NOS systems by host node number. The default NPI IDs are C0(16) and C1(16).

CDCNET_PROTOCOL_ID (CPI)

Specifies one or more protocol IDs to identify outcalls that are to be routed to NOS/VE systems or other gateways by application title. The default CDCNET protocol ID is C2(16).

NOS_DHOST_NUMBER (NDN)

Specifies 1 through 15 destination host (DHOST) numbers for the host supported by this gateway. Each NDN consists of two hexadecimal digits. The digits are used to construct the second type of title for calls to this NOS host from other NOS hosts. NOS applications access this NOS host by constructing OUTCALL blocks with a DHOST field value equal to the digits defined by this parameter. The actual titles registered in the directory are in the format *$GW_NP_xx*, where the *xx* portion consists of the ASCII equivalent of the hexadecimal digits. If no NDN value is specified, a default title will be constructed from the coupler node number received from the host when the gateway's connection to the host is opened. That is, the default title is *$GW_NP_cn*, where *cn* is the ASCII equivalent of the coupler node value.

TITLE or TITLES (T)

Specifies the third type of title or titles by which this gateway can be accessed. This title is used to support calls to the connected NOS host that originate in a system other than another NOS host. There is no default value.

DEFAULT_TRANSLATION_DOMAIN (DTD)

Specifies the portion of the catenet to which the services of this gateway are to be made available. The default value is CATENET. For this release of CDCNET, the only supported value is CATENET.

DEFAULT_SEARCH_DOMAIN (DSD)

Specifies the portion of the catenet that should be searched for the service corresponding to the title information received by the gateway in ICN/AP/R messages. The default value is CATENET. For this release of CDCNET, the only supported value is CATENET.

START (S)

Specifies whether or not the NP gateway should be started after it is configured. The default value is TRUE; started. Currently, the START parameter is ignored. Its value will always be TRUE, even if FALSE is specified on the command.

Responses

NP gateway <name> is defined.

NP gateway <name> is defined and started.

--WARNING-- NP gateway <name> is defined for trunk <name>. NP interface for the trunk is started but the logical link is down.

--WARNING-- NP gateway <name> is defined for trunk <name>. NP interface for the trunk is not started. Start NP interface to enable NP gateway.

--ERROR-- NP gateway <name> is already defined.

--ERROR-- NP gateway is already defined for trunk <name>.

--ERROR-- NP interface is not defined for trunk <name>.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--ERROR-- Protocol ID <protocol_id> already assigned.

--FATAL-- NP gateway cannot be defined. Unable to initialize the NP gateway function.

--FATAL-- NP gateway cannot be defined.
Not enough memory is currently available for required table space.

Examples

```
define_np_gw nos_dhost_number=0A1(16)
```

DEFINE_NP_INTERFACE (DEFNI)

Purpose

Defines the network block protocol interface program (BIP) to the NOS host. BIP is a software component that helps CDCNET applications and gateways connect to Network Access Method (NAM) in a NOS host.

This command is only needed in MDIs with more than one MCI, or when you choose to have and MDI be loaded over an Ethernet network solution.

An "unable to start" error message indicates that the interface was defined but not started. The COUPLER_NODE and MDI_NODE parameters will only need to be specified in a system configuration procedure if the host is installed with a version of software which does not support the coupler node verification feature, and the MDI is not going to be loaded over the channel. A DEFINE_NP_INTERFACE command specifying the COUPLER_NODE and MDI_NODE parameters should not be used in any other case.

Format

DEFINE_NP_INTERFACE
TRUNK_NAME = name
MDI_NODE = 0..FF(16)
COUPLER_NODE = 0..FF(16)
INTERFACE_NAME = name
CONGESTED_THRESHOLD = 20..255
START = boolean

Parameters

TRUNK_NAME (TN)

Name of the channel trunk to be used for this interface. The channel trunk with this name must be configured prior to execution of this command.

MDI_NODE (MN)

MDI node identifier of the logical link. MN must be set equal to the NT parameter on the NOS host's EST definition for this logical link and to the DNODE parameter on OUTCALL statements for outcalls to be carried over this logical link. There is no default for this optional parameter.

COUPLER_NODE (CN)

Coupler node identifier of the logical link. This parameter must be set equal to the ND parameter on the NOS host's EST definition for this logical link. If this parameter is omitted, the coupler node number will be obtained from the host.

INTERFACE_NAME (IN)

Logical name of the NP interface. The default name is the trunk name.

CONGESTED_THRESHOLD (CT)

Specifies the number of messages in the Block Interface Protocol (BIP) outgoing queue at which the network products interface is considered congested. The default value is 30 messages. The point at which the NP interface is again considered uncongested is 75 percent of the congested threshold.

START (S)

Specifies whether or not the NP interface should be started. The default value is TRUE; started.

Responses

NP interface <interface_name> is defined.

NP interface <interface_name> is defined and started.

--WARNING-- NP interface <interface_name> command processor has timed out waiting for a response from the NP interface.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not a Channel trunk.

--ERROR-- NP interface name <interface_name> is already defined.

--ERROR-- NP interface <interface_name> is already defined for trunk <trunk_name>.

--ERROR-- Trunk <name> already assigned.

--ERROR-- Coupler node <xxx> already assigned.

--FATAL-- NP interface cannot be defined.
Not enough memory is currently available for required table space.

--FATAL-- Unable to start NP interface <interface_name>.
Unable to start task SVM.

--FATAL-- Unable to start NP interface <interface_name>.
Unable to start task BIP.

--FATAL-- Unable to start NP interface <interface_name>.
Unable to send ITM to NP interface task.

--FATAL-- Unable to start NP interface <interface_name>.
Memory management SAP table not found.

--FATAL-- Unable to start NP interface <interface_name>.
Unknown status returned from open memory SAP.

Examples

```
define_np_interface trunk_name=$mci3
```

DEFINE_NP_TERMINAL_GW (DEFNTG)

Purpose

Defines the CDCNET titles by which the host connected to this MDI is to be known when the host is being accessed for a terminal-to-application (T-A) interactive connection or a terminal-to-Remote Batch Facility (T-RBF) connection. These titles are specified as 1- to 31-character logical identifiers of the NOS host.

This command also defines the trunk between the NOS host and the MDI to be used to support T-A interactive and T-RBF batch traffic. If the TRUNK_NAME parameter is omitted, the default channel trunk from the DEFINE_SYSTEM command will be used.

Format

DEFINE_NP_TERMINAL_GW
TITLE = list 1..15 of name
GATEWAY_NAME = name
TRUNK_NAME = name
TRANSLATION_DOMAIN = name
DEFAULT_TERMINAL_CLASS = 1..8 or 18
TERMINAL_MODEL_MAPPING = list 1..32 of (string 1..25, integer 1..18)
BATCH_TITLE = list 1..15 of name
DEFAULT_BATCH_TERMINAL_CLASS = keyword value
START = boolean

Parameters

TITLE (T)

Specifies the titles by which the host associated with this MDI is to be known by interactive terminal users accessing CDCNET.

GATEWAY_NAME (GN)

Logical name of the gateway used in subsequent commands that reference the gateway. If GATEWAY_NAME is not specified, the TRUNK_NAME parameter value will be assigned as the default gateway name.

TRUNK_NAME (TN)

Specifies the trunk name of the NOS host/MDI logical link which is to be used to support the interactive T-A traffic or batch T-RBF traffic with this host. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified via a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk. If the MDI/MTI was not loaded over a channel, and no default channel trunk was specified on a the DEFINE_SYSTEM command, then no default channel trunk exists.

TRANSLATION_DOMAIN (TD)

Specifies the portion of the catenet to which the services of this gateway are to be made available. The default value is CATENET. For this release of CDCNET, the only valid entry is CATENET.

DEFAULT_TERMINAL_CLASS (DTC)

Specifies the terminal class to be supplied by the gateway in the terminal connection request (ICN/TE/R) message sent to NAM. This value is used when the terminal class cannot be determined from the terminal model mapping pairs. The default value is 3.

TERMINAL_MODEL_MAPPING (TMM)

Specifies a list of pairings between terminal models (string 1..25) and Network Products terminal classes (1..18). The gateway references the list, using the terminal model to find out which terminal class to use in an ICN/TE/R message. A default pairing list is provided by the NP Terminal Gateway, shown in the following table.

If a terminal model in the default list is redefined by a DEFNTG command, the DEFNTG defined pairing replaces the pairing from the default list. A DEFNTG definition of a new terminal model adds the new pairing to the pairing list.

If no terminal model is specified by a user or if the specified terminal model is not found in the following table, the default terminal class will be used.

Terminal Model	Class	Manufacturer
cdc_721	3	CDC
cdc721	3	CDC
cdc_722	2	CDC
cdc722	2	CDC
cdc722_30	2	CDC
cdc_722_30	2	CDC
mac_connect_10	7	Macintosh/Connect 1.0
pc_connect_10	7	IBM PC/Connect 1.0
pc_connect_11	7	IBM PC/Connect 1.1
pc_connect_12	7	IBM PC/Connect 1.2
dec_vt100	7	Digital Equipment Corp.
vt100	7	Digital Equipment Corp.
dec_vt220	7	Digital Equipment Corp.
ibm_hasp_post	9	IBM (HASP post-print)
ibm_hasp_pre	14	IBM (HASP pre-print)
ibm_3270	18	IBM

BATCH_TITLE or BATCH_TITLES (BT)

Specifies the titles by which the host associated with this MDI is to be known by Remote Batch Facility users accessing the host through CDCNET. If BT is not specified, RBF access is not supported by this gateway definition.

DEFAULT_BATCH_TERMINAL_CLASS (DBTC)

Specifies the default terminal class for batch devices. Valid entries include the following:

Keyword Value	Description
9	HASP postprint
14	HASP preprint
18	3270 BSC

Default value is 9.

START (S)

Specifies whether or not the NP terminal gateway should be started. The default value is TRUE; started. Currently, the START parameter is ignored. Its value will always be TRUE, even if FALSE is specified on the command.

Responses

NP terminal gateway <name> is defined.

NP terminal gateway <name> is defined and started.

--WARNING-- NP terminal gateway <name> is defined for trunk <name>. NP interface for the trunk is started but the logical link is down.

--WARNING-- NP terminal gateway <name> is defined for trunk <name>. NP interface for the trunk is not started. Start NP interface to enable NP terminal gateway.

--ERROR-- NP terminal gateway <name> is already defined.

--ERROR-- NP terminal gateway is already defined for trunk <name>.

--ERROR-- NP interface is not defined for trunk <name>.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--ERROR-- Improper Terminal Model Mapping element <xx> specified. The first element in the set must be STRING (31) or NAME, the second element must be an INTEGER value of 1 to 8.

--ERROR-- Invalid DBTC parameter value <xx> was specified.

--ERROR-- Invalid DTC parameter value <xx> was specified.

--FATAL-- NP terminal gateway cannot be defined. Unable to initialize the NP terminal gateway function.

--FATAL-- NP terminal gateway cannot be defined.
Not enough memory is currently available for required table space.

Examples

```
define_np_terminal_gw title=ARHSES batch_title=RBFBATCH
```

DEFINE_OPERATOR_SUPPORT (DEFOS)

Purpose

Defines and starts the Operator Support Application in this MDI or MTI to allow network operators to communicate with the network DIs through this MDI or MTI, using the Network Operator Utility (NETOU). NETOU must be configured and running on the NOS host to which the MDI is connected.

The operator support for each unique trunk name operates independently. That is, the operator support for multiple trunk names operates as if the trunk names were defined on separate MDI/MTI systems. Commands addressed to the Operator Support Application function for one trunk will not affect the Operator Support Application function for other trunks.

Format

DEFINE_OPERATOR_SUPPORT
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

The trunk name of the logical link which is to be used for the operator support connection. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified via a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk. If the MDI/MTI was not loaded over a channel, and no default channel trunk was specified on a the DEFINE_SYSTEM command, then no default channel trunk exists.

Responses

Operator Support is defined for trunk <name>.

--WARNING-- Operator Support is defined for trunk <name>. NP Interface for the trunk is started but the logical link is down.

--WARNING-- Operator Support is defined for trunk <name>. NP interface for the trunk is not started. Start NP interface to enable Operator Support.

--ERROR-- NP Interface is not defined for trunk <name>.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--ERROR-- Operator Support for trunk <name> is already defined.

--FATAL-- Operator Support cannot be defined for trunk <name>. Not enough memory is currently available for required table space.

--FATAL-- Operator Support cannot be defined for trunk <name>. Unable to initialize the Operator Support function.

Remarks

This command is required only in the configuration files of NOS MDIs or MTIs that are selected by the site to support an operator interface to the channel-connected NOS host. It should not be placed in the configuration procedures of MDIs connected to NOS/VE hosts. The connected NOS host must be configured to run the Network Operator Utility (NETOU).

Examples

```
define_operator_support
```

DEFINE_RECORDER_LOG_GROUP (DEFRLG)

Purpose

Defines the log recorder function in the MDI or MTI that you are configuring. Specifies the name of the log group that this MDI or MTI supports, and the priority for the log recording support. This command is required only in the system configuration files of MDIs or MTIs that you select to support a logging interface to the CDC host, using the Independent Log ME. The host must be configured to run the Network Log Server application, NETLS.

The recorder log groups for each unique trunk name will operate independently. That is, the recorder log groups for multiple trunk names operate as if the trunk names were defined on separate MDI/MTI systems. Commands addressed to the Independent Log ME function for one trunk will not effect the Independent Log ME function for other trunks. For CDCNET 1.2.5, only one recorder log group can be configured per channel trunk for each MDI/MTI system.

Format

DEFINE_RECORDER_LOG_GROUP
LOG_GROUP = list range 1..1 of name or keyword value
PRIORITY = list of 1..0FF(16)
TRUNK_NAME = name

Parameters

LOG_GROUP (LG)

Name of the log group for which this log recorder is to record log messages. The default log group name is CATENET (all the DIs in the catenet). A DI can belong to only one log group.

PRIORITY (P)

Priority at which the log group is to be supported. The highest priority and default value is 1.

TRUNK_NAME (TN)

The trunk name of the logical link which is to be used for the connection to the Network Log Server application on the NOS host . If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified via a DEFINE_SYSTEM command. If a default channel trunk is not specified on the DEFINE_SYSTEM command, the channel over which the MDI/MTI was loaded is the default channel trunk. If the MDI/MTI was not loaded over a channel, and no default channel trunk was specified on a the DEFINE_SYSTEM command, then no default channel trunk exists.

Responses

Recorder log group is defined for trunk <name>.

--WARNING-- Recorder log group is defined for trunk <name>. NP interface for the trunk is started but the logical link is down.

--WARNING-- Recorder log group is defined for trunk <name>. NP interface for the trunk is not started. Start NP interface to enable log recording.

--ERROR-- NP interface is not defined for trunk <name>.

--ERROR-- No default channel trunk is defined. A trunk name must be specified.

--ERROR-- A recorder log group is already defined for trunk <name>.

--FATAL-- Recorder log groups cannot be defined for trunk <name>. Not enough memory is currently available for required table space.

--FATAL-- Recorder log groups cannot be defined for trunk <name>. Unable to initialize the log recording function.

Examples

Two MDIs in a network are to be configured with the logging recorder function. MDI_1's logging recorder has the highest priority (1). MDI_2's logging recorder has a priority of 2. If MDI_1 becomes unavailable, transmission of all log messages will be switched to MDI_2. The following commands would be used in the configuration procedures for MDI_1 and MDI_2 to configure them with the logging recorder function.

MDI_1's configuration procedure:

```
define_recorder_log_group priority=1
```

MDI_2's configuration procedure:

```
define_recorder_log_group priority=2
```

Terminal Definition Procedure Commands

This section contains descriptions of commands used in terminal definition procedures (TDPs). These commands can be executed only by executing a TDP containing the commands. These commands cannot be entered individually through the Network Operator Utility (NETOU) or by the terminal user.

The following commands are described in this section:

DEFINE_ACCESSIBLE_REMOTE_SYSTEM (DEFARS)
DEFINE_BATCH_DEVICE (DEFBD)
DEFINE_BATCH_STREAM (DEFBS)
DEFINE_I_O_STATION (DEFIOS)
DEFINE_NP_BATCH_STATION (DEFNBS) (NOS Only)
DEFINE_REMOTE_SYSTEM (DEFRS)
DEFINE_TERMINAL_DEVICE (DEFTD)
DEFINE_USER_I_O_STATION (DEFUIOS)

DEFINE_ACCESSIBLE_REMOTE_SYSTEM (DEFARS)

Purpose

Defines a Network Transfer Facility (NTF) remote system that is accessible via the directly connected NTF remote system defined by the DEFINE_REMOTE_SYSTEM command. This command may be executed only by inclusion in a TDP.

Format

DEFINE_ACCESSIBLE_REMOTE_SYSTEM
ACCESSIBLE_REMOTE_SYSTEM_NAME = name
LINE_NAME = name
AUTHORITY_LEVEL = keyword value

Parameters

ACCESSIBLE_REMOTE_SYSTEM_NAME (ARSN)

Specifies the logical name of a remote system which is accessed through the directly connected remote system.

LINE_NAME (LN)

Specifies the logical name of the line connected to the directly connected remote system through which the accessible remote system is reached. If line name does not match the line name specified on the DEFINE_LINE command, the DEFARS command is ignored. If this parameter is not specified, the line name defaults to the name of the activating line.

AUTHORITY_LEVEL (AL)

Specifies the authority level assigned to the accessible remote system being defined. The following keyword values are allowed for this parameter:

Keyword Value	Description
NET	Specifies that remote system operators are allowed to modify the logical configuration of the NTF network, as well as status and control files on the local NOS/VE system.
JOB	Specifies that remote system operators are allowed to status and control files on the local NOS/VE system.
NONE	Specifies that there is no authority level (no authorization) at the remote system.

Default is NONE. This parameter is used for NJE remote systems, and is ignored for HASP remote systems.

Responses

Accessible remote system xxx is defined.

--ERROR-- Directly connected remote system <name> may not be defined as an accessible remote system.

--ERROR-- A define_accessible_remote_system command may only be used by lines serviced by the NTF TIP.

--ERROR-- A define_accessible_remote_system command may not be included in a Terminal Definition Procedure executed via a DO command.

--ERROR-- A define_remote_system command must precede the first define_accessible_remote_system command in a Terminal Definition Procedure.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_accessible_remote_system ..
  accessible_remote_system_name=NODE_C
```

DEFINE_BATCH_DEVICE (DEFBD)

Purpose

Defines a batch device on a configured I/O station. This command can be executed only by inclusion in a TDP. Some parameters do not apply to batch devices for exclusive NOS use. The NOS/VE and NOS formats are listed separately.

Configuring a device as a batch device implies that I/O for the device is supported through the Batch Transfer Protocol (BTP). This command allows a device to be connected through the Batch Transfer Facility (BTF). To configure a device for connections through Virtual Terminal Protocol (VTP), use DEFINE_TERMINAL_DEVICE.

Devices supported by the XPCTIP can be configured only by DEFINE_TERMINAL_DEVICE. Do not use DEFINE_BATCH_DEVICE to configure devices supported by XPCTIP.

For CDC 585 printers supported by NOS Printer Support Utility (PSU), some parameters are controlled by PSU rather than DEFINE_BATCH_DEVICE. Refer to the Remarks section for more information.

Format

NOS/VE Format:

DEFINE_BATCH_DEVICE
DEVICE_NAME = name
LINE_NAME = name
CLUSTER_ADDRESS = 0..255
DEVICE_ADDRESS = 0..255
DEVICE_TYPE = keyword value
BANNER_HIGHLIGHT_FIELD = keyword value
BANNER_PAGE_COUNT = 0..3
CARRIAGE_CONTROL_SUPPORT = keyword value
DEVICE_ALIAS_1 = name
DEVICE_ALIAS_2 = name
DEVICE_ALIAS_3 = name
EXTERNAL_CHARACTERISTICS_1 = string 1..6
EXTERNAL_CHARACTERISTICS_2 = string 1..6
EXTERNAL_CHARACTERISTICS_3 = string 1..6
EXTERNAL_CHARACTERISTICS_4 = string 1..6
FILE_ACKNOWLEDGEMENT= boolean
FORMS_CODE_1 = string 1..6
FORMS_CODE_2 = string 1..6
FORMS_CODE_3 = string 1..6
FORMS_CODE_4 = string 1..6
MAXIMUM_FILE_SIZE = integer
PAGE_LENGTH = 0..128
PAGE_WIDTH = 1..255
TERMINAL_MODEL = name
TRANSMISSION_BLOCK_SIZE = 128..4095
CODE_SET = keyword value
VFU_LOAD_PROCEDURE = name
VERTICAL_PRINT_DENSITY = keyword value
FORMS_SIZE = string 1..4
UNDEFINED_FE_ACTION = keyword value
UNSUPPORTED_FE_ACTION = keyword value
VFU_LOAD_OPTION = keyword value

Format **NOS Format:**

DEFINE_BATCH_DEVICE
DEVICE_NAME = name
LINE_NAME = name
CLUSTER_ADDRESS = 0..255
DEVICE_ADDRESS = 0..255
DEVICE_TYPE = keyword value
CARRIAGE_CONTROL_SUPPORT = keyword value
EXTERNAL_CHARACTERISTICS_1 = string 1..6
EXTERNAL_CHARACTERISTICS_2 = string 1..6
EXTERNAL_CHARACTERISTICS_3 = string 1..6
EXTERNAL_CHARACTERISTICS_4 = string 1..6
PAGE_LENGTH = 0..128
PAGE_WIDTH = 1..255
TERMINAL_MODEL = name
TRANSMISSION_BLOCK_SIZE = 128..4095
CODE_SET = keyword value
VFU_LOAD_PROCEDURE = name
VERTICAL_PRINT_DENSITY = keyword value
FORMS_SIZE = string 1..4
UNDEFINED_FE_ACTION = keyword value
UNSUPPORTED_FE_ACTION = keyword value
VFU_LOAD_OPTION = keyword value

Parameters **DEVICE_NAME (DN)**

Specifies the logical name of the batch device. For batch devices to be connected to NOS Remote Batch Facility (RBF), the device names must be of the form *x...xn*, where *x...x* is any string of one to six characters, excluding underscore (_), and *n* is a digit in the range of 1..7. The value of *n* is the device ordinal for RBF use. If the DEVICE_NAME parameter value is longer than seven characters, the first underscore (_) in the first eight characters is taken as a delimiter such that the NOS RBF name for the device is the characters preceding the underscore. For example, LINE1_PRINTER would have a NOS RBF name of LINE1. If no underscore is within the first eight characters, the first seven characters are the RBF name. For example, LINEPR1_96 would have an RBF name of LINEPR1. Example device names that allow RBF connection are LP1, LP2, READER1, PUNCH1 and PRINTR5.

LINE_NAME (LN)

Specifies the logical name of the line to which the device is attached. If a DEFBD command is part of a TDP executed by the activation of a line (by a DEFL reference), LINE_NAME defaults to the name of the activating line. If a DEFBD command is part of a TDP executed by a DO command, LINE_NAME defaults to the name of the terminal user's line. Only DEFBD commands whose line names match or default to the activating line or terminal user's line are executed.

CLUSTER_ADDRESS (CA)

Specifies the cluster address to be used by the TIP for communication with the batch device. This parameter is used by the HASP, BSC3270, and MODE4 TIPs only. The default value is the default value specified on the DEFINE_TIP command, which is 0, unless a particular TIP sets it otherwise. For the HASP TIP, only one cluster is allowed on each line. For the MODE4 TIP, the cluster address must be in the range of 70(16) through 7F(16) for Mode 4A clusters, and 20(16) through 7F(16) for Mode 4C clusters. The default cluster address for MODE4 TIP is 70(16).

DEVICE_ADDRESS (DA)

Specifies the device address to be used by the TIP for communication with the batch device. This parameter is used by the HASP, BSC3270, and MODE4 TIPs only.

For devices supported by the HASP TIP, the DEVICE_ADDRESS parameter is ignored if DEVICE_TYPE=CONSOLE, and need not be specified. Only one console is allowed per cluster or line. All other device types must have a device address ranging from 1 through 7, corresponding to the stream number of the HASP workstation device being configured. If not specified, the DA parameter will default to 1 for HASP batch devices.

For the MODE4 TIP, the DEVICE_ADDRESS parameter must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. The default device address for the MODE4 TIP is 61(16).

The default device address is TIP-dependent. The DA parameter normally defaults to 0 unless a particular TIP sets it otherwise.

DEVICE_TYPE (DT)

Specifies the type of batch device. The following device types are supported.

READER (R)
PRINTER (PR)
PUNCH (PU)
PLOTTER (PL)

The default device type is PRINTER. The TIP must be able to support the specified device type; otherwise this command is rejected. For example, an error will be reported if DEVICE_TYPE is anything other than PRINTER for the URI TIP.

The following table shows the batch device types supported by each TIP:

TIP	Batch Device Types Supported
ASYNC	PRINTER
BSC3270	PRINTER
HASP	PRINTER, PLOTTER, READER, PUNCH
MODE4	READER, PRINTER
BSCNJEF	None
URI	PRINTER
XPC	None
NTF	None

BANNER_HIGHLIGHT_FIELD (BHF)

Specifies which of the banner fields is to be given prominence for files output on this device. BANNER_HIGHLIGHT_FIELD is appropriate for PRINTER and PUNCH devices only. The following parameter values are supported.

COMMENT_BANNER (CB)
ROUTING_BANNER (RB)
SITE_BANNER (SB)
USER_FILE_NAME (UFN)
USER_NAME (UN)

The default banner highlight field is ROUTING_BANNER.

The actual text in these banner highlight fields is defined by other commands.

On NOS/VE, the COMMENT_BANNER and ROUTING_BANNER text are defined by the NOS/VE user command CHANGE_JOB_ATTRIBUTES:

```
CHANGE_JOB_ATTRIBUTES COMMENT_BANNER=<comment banner text>
CHANGE_JOB_ATTRIBUTES ROUTING_BANNER=<routing banner text>
```

The SITE_BANNER text is initially defined by the CHANGE_JOB_ATTRIBUTE_DEFAULTS command, which can only be entered from the NOS/VE system console:

```
CHANGE_JOB_ATTRIBUTE_DEFAULTS SITE_INFORMATION=site banner text
```

After configuration, the banner highlight field can be changed by the CHANGE_BATCH_DEVICE_ATTRIBUTES subcommand in the NOS/VE OPERATE_STATION utility.

BANNER_PAGE_COUNT (BPC)

Specifies the number of copies of banner pages that this device is to include with files output on this device. BPC is appropriate for printer and punch devices only. If the banner page count is set to 0, no accounting information will be sent to a printer following an output file. The default banner page count is 1.

CARRIAGE_CONTROL_SUPPORT (CCS)

Specifies the types of carriage control actions that the device supports. CCS is appropriate for printer devices only. The following keyword values are allowed.

PRE_PRINT (PRE)
POST_PRINT (POST)
BOTH (B)

The default carriage control support is BOTH. This parameter is ignored by the Asynchronous and URI TIPs.

DEVICE_ALIAS_1 (DA1)

Specifies the first alternative name by which the device can be referenced. The same device alias name can be assigned to more than one device in an I/O station.

DEVICE_ALIAS_2 (DA2)

Specifies the second alternative name by which the device can be referenced. The same device alias name can be assigned to more than one device in an I/O station.

DEVICE_ALIAS_3 (DA3)

Specifies the third alternative name by which the device can be referenced. The same device alias name can be assigned to more than one device in an I/O station.

EXTERNAL_CHARACTERISTICS_1 or EXT_CHARACTERISTICS_1 (EC1)

Specifies the first external device characteristic string supported by this device. External characteristics may specify, for example, the train type of a printer device (such as A6 for uppercase and A9 for uppercase and lowercase ASCII); the name of a plotter or the plotter's manufacturer; or the default code set for a card reader (such as 026 or 029). For a PRINTER device, the default value for this parameter is NORMAL. For any other device type, the default for this parameter is to define no external characteristics. For card reader batch devices and for NOS batch devices, only the EXTERNAL_CHARACTERISTICS_1 parameter has meaning.

NOS RBF supports the following printer train types:

B6, A6, A9

and the following plotter types:

TR6, TR8

NOS PSU supports the following printer train types:

B6, A6, A9 (PSU treats B6 and A6 as the same.)

The parameter name EXT_CHARACTERISTICS_1 will be removed in a future release. For compatability, use EXTERNAL_CHARACTERISTICS_1 or EC1 instead.

EXTERNAL_CHARACTERISTICS_2 or EXT_CHARACTERISTICS_2 (EC2)

Specifies the second external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

The parameter name EXT_CHARACTERISTICS_2 will be removed in a future release. For compatability, use EXTERNAL_CHARACTERISTICS_2 or EC2 instead.

EXTERNAL_CHARACTERISTICS_3 or EXT_CHARACTERISTICS_3 (EC3)

Specifies the third external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

The parameter name EXT_CHARACTERISTICS_3 will be removed in a future release. For compatability, use EXTERNAL_CHARACTERISTICS_3 or EC3 instead.

EXTERNAL_CHARACTERISTICS_4 or EXT_CHARACTERISTICS_4 (EC4)

Specifies the fourth external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

The parameter name EXT_CHARACTERISTICS_4 will be removed in a future release. For compatability, use EXTERNAL_CHARACTERISTICS_4 or EC4 instead.

FILE_ACKNOWLEDGEMENT (FA)

Specifies whether or not file acknowledgement messages related to the device are to be displayed on the station operator's console. Default is the FILE_ACKNOWLEDGEMENT parameter value from the DEFIOS or DEFUIOS command for the I/O station to which the device belongs. If the value for the DEFIOS or DEFUIOS FILE_ACKNOWLEDGEMENT parameter is YES, file acknowledgement may not be set to NO for individual devices for that I/O station.

FORMS_CODE_1 (FC1)

Specifies the first forms code string supported by the device. FORMS_CODE_1 is appropriate for PRINTER device types, only. Forms codes are used to select the files that may be printed on the device. The default for FORMS_CODE_1 is NORMAL. This parameter and other FORMS_CODE parameters are not used for NOS batch devices.

FORMS_CODE_2 (FC2)

Specifies the second forms code string supported by the device. FORMS_CODE_2 is appropriate for PRINTER device types, only. Forms codes are used to select the files that may be printed on the device.

FORMS_CODE_3 (FC3)

Specifies the third forms code string supported by the device. FORMS_CODE_3 is appropriate for PRINTER device types, only. Forms codes are used to select the files that may be printed on the device.

FORMS_CODE_4 (FC4)

Specifies the fourth forms code string supported by the device. FORMS_CODE_4 is appropriate for PRINTER device types, only. Forms codes are used to select the files that may be printed on the device.

MAXIMUM_FILE_SIZE (MFS)

Specifies the maximum size in bytes of any file that may be output to the device. MAXIMUM_FILE_SIZE is appropriate for PRINTER, PLOTTER, or PUNCH device types, only. If MFS is not specified, no file size limit is defined.

PAGE_LENGTH (PL)

Specifies the number of output lines that constitute a page for this device. In CDCNET 1.2.5, this parameter is being replaced by the FORMS_SIZE parameter. For compatibility, this parameter is allowed on the DEFINE_BATCH_DEVICE command, but will be ignored.

PAGE_WIDTH (PW)

Specifies the number of columns that constitute a line for this device. PAGE_WIDTH is appropriate for PRINTER, PUNCH, and PLOTTER device types. The default page width for PRINTER devices is 136 columns. The default page width for PUNCH or PLOTTER device types is 80 columns.

TERMINAL_MODEL (TM)

Specifies a 1- to 25-character terminal model name for the device. For READER, PUNCH and PLOTTER device types, TERMINAL_MODEL is informational only, and may be set to any user-defined model name. For PRINTER devices, TERMINAL_MODEL defines the printer attributes supported by the printer. The following default values are defined for the TIPs supported by CDCNET.

TIP	Default Printer Attributes
ASYNC	C536
HASP	C18
URI	C585V
MODE4	M4IMP

Other CDC-supplied values for printers that you may specify for this parameter include the following:

C537 (for a CDC 537 printer)

M4NIMP (for a Mode 4 non-impact printer)

These printer model names are not TIP-defined default values. To reference such printer models, you must specify the TM parameter on the DEFBD command.

For more information on printer attributes, see the description of the DEFINE_PRINTER_MODEL_ATTRIBUTES command in this chapter.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the TIP for communication with the batch device. The value on this command overrides the TRANSMISSION_BLOCK_SIZE parameter on the DEFINE_LINE command for this device only. The default value is the value specified on the DEFINE_LINE command. This parameter is ignored by the URI TIP.

CODE_SET (CS)

Indicates if the TIP has to map characters from ASCII-128 to the printer code set. The following values are allowed:

ASCII
ASCII48
ASCII64
ASCII95
ASCII128
EBCDIC

The value ASCII implies no character mapping is required.

The default code set depends upon the TIP supporting the batch device:

TIP	Default Code Set
ASYNC	ASCII128 (Only allowed code set)
HASP	EBCDIC (Only allowed code set)
URI	ASCII95
MODE4	ASCII95 (ASCII64 is also supported for Mode 4 BCD terminals.)

EBCDIC is not supported by the ASYNC and URI TIPs.

VFU_LOAD_PROCEDURE (VLP)

Specifies the name of the procedure containing the default VFU load image for the batch device. The default VFU load procedure, if the value for the VFU_LOAD_OPTION is any value but NONE, is the CDC-supplied procedure CDC_VFU.

The VLP is executed when a printer initially becomes active. Although the resultant VFU load image is loaded into the printer every time the line is started after being stopped, the DEFINE_VFU_LOAD_IMAGE (DEFVLI) commands in the VLP are guaranteed to be reprocessed *only* when a CHANGE_BATCH_DEVICE_ATTRIBUTES (CHABDA) command that specifies the VLP parameter is executed. Other changes and conditions during normal processing may also cause the DEFVLI commands to be reprocessed.

VERTICAL_PRINT_DENSITY (VPD)

Indicates the default vertical print density for the device and whether the density for the device can be changed by the TIP. The following values are allowed:

Keywords	Description
SIX_ANY, EIGHT_ANY	The vertical print density can be changed to either 6 lines-per-inch or 8 lines-per-inch.
SIX_ONLY, EIGHT_ONLY	The vertical print density cannot be changed.

The default for Asynchronous, HASP, and MODE4 TIPs is SIX_ONLY. The default for the URI TIP is SIX_ANY.

FORMS_SIZE (FS)

A string value that represents the length, in inches, of the forms in the printer. Strings representing decimal numbers that are multiples of half inches from .5 to 31 inches are allowed. This parameter replaces the PAGE_LENGTH parameter. The forms size value will be passed to the control facility, and will be used, with the file-specified vertical print density, to select files for printing. The default forms size is 11.

UNDEFINED_FE_ACTION (UNDFA) or *UN_DEFINED_FE_ACTION (UDFA)*

Indicates the action the TIP should take with format effectors that are not defined. The following values are allowed:

Keyword	Description
PAS	Print after spacing
PBS	Print before spacing
DPL	Discard print line

Default is PAS.

Parameter names UN_DEFINED_FE_ACTION and UDFA will be removed in a future release. For compatibility, use UNDEFINED_FE_ACTION or UNDFA instead.

UNSUPPORTED_FE_ACTION (UNSFA) or *UN_SUPPORTED_FE_ACTION (USFA)*

Indicates the action the TIP should take with format effectors defined but not supported by the device. The following keyword values are allowed for this parameter:

Keyword	Description
PAS	Print after spacing
PBS	Print before spacing
DPL	Discard print line

Default is DPL.

Parameter names UN_SUPPORTED_FE_ACTION and USFA will be removed in a future release. For compatibility, use UNSUPPORTED_FE_ACTION or UNSFA instead.

VFU_LOAD_OPTION (VLO)

Indicates the presence of a loadable vertical format unit (VFU) load image for the batch device, as well as any restrictions on changing the VFU load image. The following keyword values are allowed for this parameter:

Keyword	Description
NONE	VFU load image not present or not loadable.
INIT	Load VFU load image during initialization only; VFU load image cannot be changed by I/O station operator or user.
OPER	Default VFU load image can be changed by the I/O station operator.
USER	The I/O station operator can change the default VFU load image and users can change the VFU for individual files.

The default value for batch devices supported by the Asynchronous TIP is NONE. The default value for batch devices supported by the URI TIP is USER.

Responses Batch device <name> defined.

--ERROR-- A DEFIOS, DEFUIOS or DEFNBS command must precede the first DEFBD command in a Terminal Definition Procedure.

--ERROR-- Batch device_name <name> is not unique within the I/O station.

--ERROR-- Line_name <name> does not match name of the terminal user's line. Line_name, if specified, must match the terminal user's line name when a Terminal Definition Procedure is executed via a DO command.

--ERROR-- <parameter_name> may not be specified for the given device type.

--ERROR-- File_acknowledgement may not be specified as NO, FALSE or OFF for the device while the device is assigned to an I/O station with file_acknowledgement specified as YES, TRUE, or ON.

--ERROR-- <parameter_name> value is not allowed.

--ERROR-- <parameter_name> keyword is not recognized.

--ERROR-- Cannot locate the specified printer terminal model.

--FATAL-- Not enough memory currently exists for required table space.

Remarks

If a CDC 585 or batch 537 printer is to be used on NOS, it will be supported by the NOS Printer Support Utility (PSU). In this case, several printer attributes are specified by PSU commands. The values specified by PSU commands override values specified by corresponding parameters on the TDP's DEFBD command that defines the printer.

Such attributes are:

DEFBD Parameter	PSU Command
BANNER_PAGE_COUNT	BANNERS
FORMS_CODE_n	FORM
FORMS_SIZE	PRSIZE
VERTICAL_PRINT_DENSITY	PRSIZE

NOTE

Although the PSU BANNERS command does override the BPC parameter regarding how many banner pages to generate, the BPC parameter has another use that is independent of the BANNERS command. In particular, if the BPC parameter is set to 0, no accounting message (such as "TRANSFER COMPLETE - nnnnnnn LINES PRINTED") is printed at the end of the file.

Examples

```
define_batch_device device_name=pr2,device_type=printer,..
  device_address=3
```

```
define_batch_device device_name=pr1,device_type=printer,..
  forms_code_1=lined
```

DEFINE_BATCH_STREAM (DEFBS)

Purpose

Defines a Network Transfer Facility (NTF) batch stream associated with the directly connected NTF remote system defined by the DEFINE_REMOTE_SYSTEM command. Defining a batch stream implies that I/O for the stream is supported through the Batch Transfer Protocol (BTP). This command may be executed only by inclusion in a TDP.

Format

DEFINE_BATCH_STREAM
STREAM_NAME = name
STREAM_TYPE = keyword value
LINE_NAME = name
STREAM_ORDINAL = 1..7
MAXIMUM_FILE_SIZE = integer
TRANSMISSION_BLOCK_SIZE = 400..4095
PAGE_WIDTH = 10..255
TRANSPARENT_MODE = boolean
SKIP_PUNCH_COUNT = 0..9
START = boolean

Parameters

STREAM_NAME (SN)

Specifies the logical name of the batch stream.

STREAM_TYPE (ST)

Specifies the type of batch stream. The following keyword values are allowed for this parameter:

READER (RD)
PRINTER (PR)
PUNCH (PU)
PLOTTER (PL)
REMOTE_SYSTEM_INPUT (RSI)
JOB_RECEIVER (JR)
SYSOUT_RECEIVER (SR)
JOB_TRANSMITTER (JT)
SYSOUT_TRANSMITTER (ST)

LINE_NAME (LN)

Specifies the logical name of the line connected to the directly connected remote system. If the line name specified does not match the line name specified on the DEFINE_LINE command, the DEFBS command is ignored. If this parameter is not specified, line_name defaults to the name from the DEFINE_LINE command.

STREAM_ORDINAL (SO)

Specifies the stream ordinal. The NTF TIP supports up to 7 receive streams for the following stream types:

PRINTER
PUNCH
PLOTTER
REMOTE_SYSTEM_INPUT
JOB_RECEIVER (with SYSOUT_RECEIVER, up to combined total of 8)
JOB_TRANSMITTER (with Sysout Transmitter, up to combined total of 8)
SYSOUT_RECEIVER

The NTF TIP supports up to 7 transmit streams for the following stream types:

READER
JOB_TRANSMITTER (with SYSOUT_TRANSMITTER, up to a combined total of 8)
SYSOUT_TRANSMITTER

For HASP remote systems, the PUNCH, PLOTTER and REMOTE_SYSTEM_INPUT streams share the HASP punch stream. The stream_ordinal assigned to these streams must be unique, for the NTF TIP to determine which HASP punch stream is to be used for each PUNCH, PLOTTER or REMOTE_SYSTEM_INPUT stream. The default value is 1.

MAXIMUM_FILE_SIZE (MFS)

Specifies the maximum size, in bytes, of any file that may be transmitted to the stream. This parameter is for transmit-type streams, only. If a value is not specified, no file size limit is defined.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the NTF TIP for initial communication with the remote system. The value on this command overrides the TBS parameter on the DEFINE_REMOTE_SYSTEM command.

PAGE_WIDTH (PW)

Specifies the number of columns that constitute a card image for the card reader stream to a HASP remote system. The default page width is 80 columns.

TRANSPARENT_MODE (TM)

Specifies whether data received on the HASP PRINTER, PUNCH, PLOTTER, and REMOTE_SYSTEM_INPUT batch streams is to be processed as transparent or nontransparent. Default is TRUE, the HASP receive stream processes data as transparent.

SKIP_PUNCH_COUNT (SPC)

Specifies the number of cards/lines at the beginning of a HASP PUNCH or REMOTE_SYSTEM_INPUT stream to be discarded. HASP remote systems typically precede data on punch streams with banner or lace cards. Default value is zero.

START (S)

Specifies that the batch stream is automatically started when the line is activated. This means that files may be transferred or received on this stream without NTF operator intervention. Default value is TRUE.

Responses Batch stream xxx is defined.

--ERROR-- A define_batch_stream command may only be used by lines serviced by the NTF TIP.

--ERROR-- A define_batch_stream command may not be included in a Terminal Definition Procedure executed via a DO command.

--ERROR-- A define_remote_system command must precede the first define_batch_stream command in a Terminal Definition Procedure.

--ERROR-- Batch stream_name <name> is not a NJE remote system protocol type stream.

--ERROR-- Batch stream_name <name> is not a HASP remote system protocol type stream.

--ERROR-- <parameter-name> may not be specified for the given stream type.

--FATAL-- Not enough memory currently exists for required table space.

Examples

```
define_batch_stream stream_name=SYSOUT_RECV1, stream_type=SR, so=1.

define_batch_stream stream_name=LINE_PRINTER_2, st=PR, so=2.
```

DEFINE_I_O_STATION (DEFIOS)

Purpose

Defines auto-configured and operator-configured public and private I/O stations for NOS/VE and NOS use.

This command may only be executed by inclusion in a TDP.

During connection to NOS systems, only the CONTROL_FACILITY, FILE_ACKNOWLEDGEMENT and P_M_ACTION parameters are used.

Format

DEFINE_I_O_STATION
I_O_STATION_NAME = name
CONTROL_FACILITY = name
DEFAULT_JOB_DESTINATION = name
STATION_USAGE = keyword value
REQUIRED_OPERATOR_DEVICE = name
I_O_STATION_ALIAS = list 1..3 of name
DESTINATION_UNAVAILABLE_ACTION = keyword value
FILE_ACKNOWLEDGEMENT = boolean
P_M_ACTION = keyword value

Parameters

I_O_STATION_NAME (IOSN)

Specifies the logical name of the I/O station. For public and private auto-configured I/O stations, this name is used to take control of I/O stations using the OPERATE_STATION (OPES) utility (OPES,STATION_NAME=station_name). In addition, for public I/O stations, you always use the value of this parameter for the STATION parameter on the PRINT_FILE command.

CONTROL_FACILITY (CF)

Specifies the name registered by the controlling Status and Control Facility Server (SCFS) for the I/O station.

On NOS/VE, this name is defined by the CONTROL_FACILITY_NAME parameter on the ACTIVATE_SCFS NOS/VE command. The value of the CONTROL_FACILITY parameter on the DEFIOS command must match the value for the CONTROL_FACILITY_NAME parameter on the ACTIVATE_SCFS command. The default control facility name for ACTIVATE_SCFS is STATION_CONTROLLER_1. ACTIVATE_SCFS is documented in the NOS/VE Software Release Bulletin. If your site plans to have more than one control facility active in your network, be sure that the two control facilities are defined with different names. Do not use the default name for both control facilities.

On NOS, this name is defined by the BATCH_TITLE parameter on the DEFINE_NP_TERMINAL_GW command.

For private I/O stations connected to NOS/VE, you specify this control facility name for the STATION parameter on the PRINT_FILE command. Users sending files to a private I/O station must know the control facility name.

If the I/O station will connect to NOS/VE only, or to NOS and NOS/VE, the CONTROL_FACILITY parameter must match the name of a NOS/VE control facility. If the CONTROL_FACILITY parameter is set to a NOS/VE control facility name, and there is a required operator device, the I/O station operator may switch between NOS/VE and NOS. For standalone printers connected to NOS, the CONTROL_FACILITY parameter must be set to a batch gateway title. Such an I/O station cannot be switched between NOS and NOS/VE.

DEFAULT_JOB_DESTINATION (DJD)

Specifies the destination to which an input file will be sent if no destination is specified on the ROUTE_JOB command for the file or if no ROUTE_JOB command is entered for the file. A job destination is a family_name registered (in the format BTFS$family_name) by a NOS/VE host. The ROUTE_JOB command indicates the job destination for the file.

STATION_USAGE (SU)

Specifies the mode of use for the I/O station. The following keyword values are allowed.

Keyword Value	Description
PUBLIC	NOS/VE output can be routed to I/O station name. The origin of batch input for a PUBLIC I/O station is the I/O station itself.
PRIVATE	Output is routed to station operator's user name. The origin of batch input for a PRIVATE I/O station is the operator at the operator's console. If PRIVATE usage is specified, a user must log in and request control of the I/O station before the batch devices can become operational. On NOS/VE, files are routed to a PRIVATE I/O station by specifying the name of the Control Facility for the I/O station rather than the I/O station name on the PRINT_FILE command.

Default is PUBLIC.

REQUIRED_OPERATOR_DEVICE (ROD)

Specifies the device name of the only console from which a user can control the I/O station. If a required operator device is not specified for an auto-configured I/O station, a user at any console may request control of the I/O station. For an operator-configured I/O station, the station entering the DO command is the required operator device if no required operator device is specified on this command.

I_O_STATION_ALIAS (IOSA)

Specifies one to three alias names for a public I/O station. If aliases are defined for a station, files can be directed to the I/O station by the station name or by one of the alias names. The same alias can be used by more than one I/O station. In this case, a file directed to the common alias is output to the I/O station with the first available device appropriate for the file. If aliases are not specified, files can be routed to the I/O station name only.

Aliases are invalid for private I/O stations.

DESTINATION_UNAVAILABLE_ACTION

Specifies the action the DI should take if the job destination for a job is unavailable. The following keyword values are allowed:

Keyword Value	Description
DROP	The job will be read and discarded and the reading of subsequent jobs will continue if the destination is unavailable.
STOP	The input device for the job will be stopped and no more jobs will be read from the device until the destination becomes available or until the operator drops the job by entering a command.

Default is STOP.

FILE_ACKNOWLEDGEMENT (FA)

Specifies whether or not the I/O station operator is to receive acknowledgement messages at the console for each file received. Default is NO, no acknowledgement.

P_M_ACTION (PMA)

Specifies how TIPs supporting the print devices for the I/O station should process print lines containing PM (printer message) as the first two characters in the line. The following keyword values are allowed.

Keyword Value	Description
DISPLAY	The line is displayed to the station operator as a printer message. A displayed printer message causes the device assigned to the print file to stop until the operator acknowledges the message by entering a START_BATCH_DEVICE command on NOS/VE and a GO command on NOS. If no operator is controlling the I/O station, output of a print file terminates when a printer message is detected. The print file is held until the operator explicitly selects it to print.
PRINT	The line is printed using the "P" format effector.
DISCARD	The print line containing the printer message is discarded.

Default is PRINT.

Responses

IO station <name> defined.

--ERROR-- IO_STATION_NAME <name> already defined as an I/O station or remote system. Station may not be redefined in a Terminal Definition Procedure executed via a DO command.

--ERROR-- STATION_USAGE must be public for a DEFINE_I_O_STATION definition in a Terminal Definition Procedure executed via a DO command.

--ERROR-- I_O_STATION_ALIAS names may not be specified for private IO stations.

--ERROR-- <parameter_name> keyword is not recognized.

--ERROR-- DEFINE_I_O_STATION, DEFINE_USER_I_O_STATION or DEFINE_NP_BATCH_STATION commands may not intermixed in the same Terminal Definition Procedure.

--FATAL-- Not enough memory is currently available for required table space.

Examples

The following example defines a public I/O station named Station1. Station1 is to be controlled by control facility SCFS109. Acknowledgement messages are printed at the I/O station control console when files are received at the I/O station. Printer messages are to be printed. Aliases for the I/O station are REM1 and ENGBLDG.

```
define_i_o_station i_o_station_name=station1,..
  control_facility=scfs109,station_usage=public,..
  file_acknowledgement=yes,p_m_action=print,..
  i_o_station_alias=(rem1,engbldg)
```

DEFINE_NP_BATCH_STATION (DEFNBS) (NOS Only)

Purpose Defines an I/O station to be used only with NOS systems running the Remote Batch Facility (RBF). This command is used for *operator-configured private I/O stations* only. It can only be executed by inclusion in a TDP that is executed by a DO command. You cannot use this command in a TDP that is specified on a DEFINE_LINE command. The control facility for the NP (Network Products) batch station is the NOS application (RBF or PSU) which supports the station.

Format **DEFINE_NP_BATCH_STATION**

Responses IO station <name> defined.

--ERROR-- Only one DEFNBS defined I/O station may be defined in a Terminal Definition Procedure executed via a DO command.

--ERROR-- DEFINE_I_O_STATION, DEFINE_USER_I_O_STATION or DEFINE_NP_BATCH_STATION commands may not be intermixed in the same Terminal Definition Procedure.

--FATAL-- Not enough memory is currently available for required table space.

Remarks DEFINE_NP_BATCH_STATION generates a name for the I/O station using a combination of the following values:

The string $IOSTATION_.

The last six hexadecimal digits of the DI's system ID.

A 4-digit decimal number in the range of 0000 through 9999. The DI software assigns this number consecutively for each $IOSTATION specification encountered. The first number assigned is 0000 (0000 follows 9999 thereafter).

For example, an I/O station connected to a DI system of ID 0800251FE029 that has last assigned number 1234 is named $IOSTATION_1FE029_1235.

Examples

```
define_np_batch_station
```

DEFINE_REMOTE_SYSTEM (DEFRS)

Purpose Defines a directly connected Network Transfer Facility (NTF) remote system. This command may only be executed by inclusion in a TDP.

Format

DEFINE_REMOTE_SYSTEM
REMOTE_SYSTEM_NAME = name
LOCAL_SYSTEM_NAME = name
CONTROL_FACILITY = name
REMOTE_SYSTEM_PROTOCOL = keyword value
LOGICAL_LINE_NUMBER = 1..999
LINE_NAME = name
AUTHORITY_LEVEL = keyword value
TERMINAL_USER_PROCEDURE = name
POSITIVE_ACKNOWLEDGE = keyword value
WAIT_A_BIT = keyword value
INACTIVITY_TIMER = 1..600 or keyword value INFINITE
REMOTE_PASSWORD = string
LOCAL_PASSWORD = string
DEFAULT_JOB_DESTINATION = name
DEFAULT_FILE_DESTINATION = name
TRANSMISSION_BLOCK_SIZE = 400..4095

Parameters

REMOTE_SYSTEM_NAME (RSN)

Specifies the logical name of the directly connected remote system.

LOCAL_SYSTEM_NAME (LSN)

Specifies the logical name used by NJE remote systems to reference the DI. This name is used in signon processing and for remote operator commands. This parameter is required for NJE remote systems, but is ignored for HASP remote systems.

CONTROL_FACILITY (CF)

Specifies the name of the Status and Control Facility Server (SCFS) which is to be the control facility for the remote system. A title is registered by the control facility as "SCFS$NTF_control_facility".

REMOTE_SYSTEM_PROTOCOL (RSP)

Specifies the mode of use for the remote system. The following keyword values are allowed for this parameter:

NJE
HASP
HASP_RBF
HASP_INTERCOM5
HASP_JES2
HASP_JES3
HASP_ASP
HASP_OTHER

If the remote system supports the HASP protocol, one of the supported HASP spoolers must be specified. If the remote system supports the NJE protocol, NJE must be specified.

LOGICAL_LINE_NUMBER (LLN)

Specifies the logical line number assigned to the line between a DI and directly connected remote system. Each remote system definition (via DEFRS commands) must have a unique line number within a control facility to allow remote operators to reference streams on specific lines.

LINE_NAME (LN)

Specifies the logical name of the line connected to the directly connected remote system. If the line name specified does not match the line name specified on the DEFINE_LINE command, the DEFRS command is ignored. If this parameter is not specified, the line name defaults to the name from the DEFINE_LINE command.

AUTHORITY_LEVEL (AL)

Specifies the authority level of the directly connected remote system operator. The following keyword values are allowed:

Keyword Value	Description
NET	Specifies that remote system operators are allowed to modify the logical configuration of the NTF network as well as status and control files on the local NOS/VE system.
JOB	Specifies that remote system operators are allowed to status and control files on the local NOS/VE system.
NONE	Specifies that there is no authority level.

Default is NONE, no authority level.

This parameter is used for NJE remote systems, but is ignored for HASP remote systems.

TERMINAL_USER_PROCEDURE (TUP)

Specifies the name of the terminal user procedure (TUP) associated with the remote system. The commands in the named TUP execute when the remote system is configured. If this parameter is not specified, no TUP is executed. This parameter is for HASP remote systems, and is ignored for NJE remote systems.

POSITIVE_ACKNOWLEDGE (PA)

Specifies what sequence should be sent to the remote system as a positive acknowledgment, if the NTF TIP has no data to transmit. The following keyword values are allowed for this parameter:

ACK
NULL

Either the ACK sequence (DLE ACK0) or a NULL block (Function Control Sequence [FCS] block) may be sent as a positive acknowledgement. During NJE signon processing, an ACK is the only valid response following receipt of an ENQ from a remote system. The NTF TIP receives either ACK or NULL block as a positive response from another system. If the remote system sends NULL blocks in place of ACK, the TIP is able to better perform block sequence error checking. The default value is ACK.

WAIT_A_BIT (WAB)

Specifies how the global wait-a-bit is cleared. The following keyword values are allowed for this parameter:

Keyword Value	Description
ACK	Receipt of an ACK clears the wait-a-bit.
FCS	The wait-a-bit is determined from the change FCS block.

The default value is ACK.

If NULL was specified for the POSITIVE_ACKNOWLEDGE parameter, then FCS is the only value which may be specified for the WAIT_A_BIT parameter. The clearing of the wait-a-bit is determined from the change FCS block.

INACTIVITY_TIMER (IT)

Specifies the amount of time (seconds) the DI waits for a response to a command sent to a HASP remote system on the console stream. If a response is not received in the specified amount of time, a message indicating no response is sent to the log file and the next available command is sent. The keyword value INFINITE disables the timer and causes the DI to wait indefinitely for a response. The default value is INFINITE. This parameter is for HASP remote systems, and is ignored for NJE remote systems.

REMOTE_PASSWORD (RP)

Specifies the password (1 through 8 characters) to be received from the NJE remote system during signon processing. Default is NULL, no password. For HASP remote systems, this parameter specifies the remote ID (1 through 999) that is included in the SIGNON sent to IBM spoolers.

LOCAL_PASSWORD (LP)

Specifies the password (1 through 8 characters) to be sent to the remote system during signon processing. Default is NULL, no password.

DEFAULT_JOB_DESTINATION (DJD)

Specifies the destination to which a HASP remote_system_input job is sent if no destination is specified on the ROUTE_JOB command for the job. For NJE remote systems, this parameter is ignored since the destination is specified by the Network Job Header. A job destination is a family_name registered (as BTFS$family_name) by a CYBER 180 NOS/VE system.

DEFAULT_FILE_DESTINATION (DFD)

Specifies the destination to which a HASP received file is sent. For NJE remote systems, this parameter is ignored since the destination is specified by the Network Job Header. A file destination is a family_name registered (as BTFS$family_name) by a CYBER 180 NOS/VE system.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the NTF TIP for initial communication with the remote system. The value on this command overrides the TBS parameter on the DEFINE_LINE command.

Responses

Remote System xxx is defined.

--ERROR-- A define_remote_system command may only be used by lines serviced by the NTF TIP.

--ERROR-- The control_facility specified for remote system <name> does not match a previously defined remote system control_facility.

--ERROR-- A define_remote_system command may not be included in a Terminal Definition Procedure executed via a DO command.

--ERROR-- Remote system name and logical line number are not unique for remote system <name>.

--ERROR-- Multiple define_remote_system commands may not be specified for the same line.

--ERROR-- Wait_a_bit may not be specified as ACK when positive_acknowledge is NULL for remote system <name>.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_remote_system ..
  remote_system_name=NODE2, local_system_name=NOS_VE2, ..
  control_facility=SCFS109, remote_system_protocol=NJE, ..
  logical_line_number=1, line_name=LINE1
```

DEFINE_TERMINAL_DEVICE (DEFTD)

Purpose Defines a terminal device on a configured line. This command executes when the communication line connected to the terminal device is activated. Except for the ASYNC TIP, CDC-provided TIPs only support the definition of CONSOLE devices via the DEFINE_TERMINAL_DEVICE command.

Configuring a device as a terminal device implies the I/O for that device is supported through the Virtual Terminal Protocol (VTP). To configure a device for connections through Batch Transfer Protocol (BTP), use DEFINE_BATCH_DEVICE. Devices supported by the XPCTIP can be configured only by DEFINE_TERMINAL_DEVICE. This command may be executed only by inclusion in a TDP.

Format **DEFINE_TERMINAL_DEVICE**
DEVICE_NAME = name
LINE_NAME = name
CLUSTER_ADDRESS = 0..255
DEVICE_ADDRESS = 0..255
DEVICE_TYPE = keyword value
TERMINAL_USER_PROCEDURE = name
TRANSMISSION_BLOCK_SIZE = 128..4095

Parameters *DEVICE_NAME (DN)*

Logical name of the terminal device.

This parameter, when specified, is also used to generate a NOS terminal name if the terminal is connected to a NOS host. If you use this parameter to define NOS terminal names, make sure that each NOS terminal name is unique throughout the network. The default value for the terminal device name is constructed using the following information:

- $ (dollar sign).
- DEVICE_TYPE parameter value from this command (default = CONSOLE).
- The system ID of the DI to which the terminal device is connected. Only the last 6 digits of the 12-hexadecimal-digit system ID are used.
- The LIM number to which the communication line supporting the terminal device is connected. This value is specified on the DEFINE_ LINE command.
- The port number to which the communication line supporting the terminal device is connected. This value is specified on the DEFINE_ LINE command.
- Cluster address for the terminal device (default = 0).
- Device address for the terminal device (default = 0).

For example, given the following information and command/parameter values:

```
system_id = 080025109999
define_line lim=4 port=2
define_terminal_device device_type=console cluster_address=00,..
terminal_address=00
```

the default device name would be $CONSOLE_109999_420000.

LINE_NAME (LN)

Logical name of the line. This parameter is optional when the DEFTD command is part of a terminal definition procedure (TDP), since the TDP is associated with a specific line and is a parameter on the DEFINE_LINE command. If this parameter is specified on a DEFINE_TERMINAL_DEVICE command within a TDP, the DEFINE_TERMINAL_DEVICE command is ignored if it does not match the LINE_NAME parameter value on the DEFINE_LINE command.

CLUSTER_ADDRESS (CA)

Cluster address to be used by the TIP for initial communication with the terminal console device. This parameter is used by the HASP, BSC3270 and MODE4 TIPs only. The default value is the default value specified on the DEFINE_TIP command, which is 0, unless a particular TIP sets it otherwise. For the HASP TIP, only one cluster is allowed on each line. The value on this command overrides the CLUSTER_ADDRESS parameter on the DEFINE_TIP command.

For the MODE4 TIP, the cluster address must be in the range of 70(16) through 7F(16) for Mode 4A clusters, and 20(16) through 7F(16) for Mode 4C clusters. The default cluster address for MODE4 TIP is 70(16).

Because the MODE4 TIP uses "cluster polling" for Mode 4C clusters, and since devices on a cluster are not auto-recognized, the configuration for each Mode 4C cluster should contain a DEFINE_TERMINAL_DEVICE command for each device in the cluster. Input from a device that is not configured results in a cluster failure.

DEVICE_ADDRESS (DA)

Device address to be used by the TIP for initial communication with the terminal console device. This parameter is used by the HASP, BSC3270 and MODE4 TIPs only. For devices supported by the HASP TIP, the DEVICE_ADDRESS parameter is ignored if DEVICE_TYPE=CONSOLE, and need not be specified.

For the MODE4 TIP, the DEVICE_ADDRESS parameter must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. The default device address for the MODE4 TIP is 61(16).

The value for this parameter on this command overrides the DEVICE_ADDRESS parameter on the DEFINE_TIP command.

For the XPCTIP, if TDPs are used to configure the terminal devices for an X.PC line (reference DEFINE_LINE command), the TDP must not contain a DEFTD command with a non-zero device address. The X.PC protocol will start only when the device address is set or defaulted to zero.

The default device address is TIP-dependent. The DA parameter normally defaults to 0 unless a particular TIP sets it otherwise.

DEVICE_TYPE (DT)

Specifies the type of terminal device. Defined parameters are: CONSOLE, READER, PRINTER, PUNCH, and PLOTTER. Default is CONSOLE. The TIP must be able to support the specified device type; otherwise the command is rejected.

The following table shows the terminal (VTP) device types supported by each TIP:

TIP	Terminal (VTP) Device Types Supported
ASYNC	CONSOLE, PRINTER
BSC3270	CONSOLE
HASP	CONSOLE
MODE4	CONSOLE
BSCNJEF	None
URI	None
XPC	CONSOLE
NTF	None

TERMINAL_USER_PROCEDURE (TUP)

Name of the terminal user procedure (TUP) to be executed for this device when the communication line supporting the device becomes active. A TUP may contain any terminal user command except ACTIVATE_AUTO_RECOGNITION. This parameter allows you to predefine a user's terminal environment and have the environment automatically set up when the line becomes active. By specifying this parameter on this command, you override the TUP parameter value on the DEFINE_TIP or DEFINE_LINE command for this device. The default TUP is the one specified on the DEFINE_TIP or DEFINE_LINE command.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the TIP for initial communication with the terminal console device. The value on this command overrides the TRANSMISSION_BLOCK_SIZE parameter on the DEFINE_TIP or DEFINE_LINE command for this device only. The default value is the value specified on the DEFINE_LINE command (if TBS is specified on that command), or the DEFINE_TIP command (if TBS is not specified on the DEFINE_LINE command).

Responses

Terminal device <device_name> defined.

--ERROR-- Line name <line_name> not defined.

--ERROR-- Parameter line_name is required, but was omitted.

--FATAL-- Not enough memory currently exists for required table space.

Examples

```
define_terminal_device device_name=trm_3,line_name=line1
```

DEFINE_USER_I_O_STATION (DEFUIOS)

Purpose

Defines an *operator-configured private* I/O station. This command is used for *operator-configured private I/O stations only*. It can be executed only by inclusion in a TDP that is executed by a DO command. You cannot use this command in a TDP that is specified on a DEFINE_LINE command. This command sets the required operator console for the station to the console entering the DO command.

Format

DEFINE_USER_I_O_STATION
CONTROL_FACILITY = name
DEFAULT_JOB_DESTINATION = name
DESTINATION_UNAVAILABLE_ACTION = keyword value
FILE_ACKNOWLEDGEMENT = boolean
P_M_ACTION = keyword value

Parameters

CONTROL_FACILITY (CF)

Specifies the name registered by the controlling Status and Control Facility Server (SCFS) for the I/O station. This name is defined by the CONTROL_FACILITY_NAME parameter on the ACTIVATE_SCFS NOS/VE command. The default control facility name for ACTIVATE_SCFS is STATION_CONTROLLER_1. ACTIVATE_SCFS is documented in the NOS/VE Software Release Bulletin. If your site plans to have more than one control facility active in your network, be sure that the two control facilities have different names. Do not use the default name for both control facilities.

For operator-configured private I/O stations, you specify this Control Facility name for the following other commands:

- On the OPERATE_STATION command. Specify the control facility name on the STATION_NAME parameter (OPERATE_STATION STATION_NAME=control_facility_name).
- On the PRINT_FILE command. Specify the control facility name on the STATION parameter (PRINT_FILE STATION=control_facility_name).

  Since users sending files to a private I/O station must know the control facility name, you should distribute the control facility name to these users.

DEFAULT_JOB_DESTINATION (DJD)

Specifies the destination to which an input file will be sent if no destination is specified on the ROUTE_JOB command for the file, or if no ROUTE_JOB command is entered for the file. A job destination is a family_name registered (in the format BTFS$family_name) by a NOS/VE host. The ROUTE_JOB command indicates the job destination for the file.

DESTINATION_UNAVAILABLE_ACTION

Specifies the action the DI should take if the job destination for a job is unavailable. The following keyword values are allowed:

Keyword Value	Description
DROP	The job will be read and discarded and the reading of subsequent jobs will continue if the destination is unavailable.
STOP	The input device for the job will be stopped and no more jobs will be read from the device until the destination becomes available or until the operator drops the job by entering a command.

Default is STOP.

FILE_ACKNOWLEDGEMENT (FA)

Specifies whether or not the I/O station operator is to receive acknowledgement messages at the console for each file received. Default is NO (no acknowledgement).

P_M_ACTION (PMA)

Specifies how TIPs supporting the print devices for the I/O station should process print lines containing PM (printer message) as the first two characters in the line. The following keyword values are allowed.

Keyword Value	Description
DISPLAY	The line is displayed to the station operator as a printer message. A displayed printer message causes the device assigned to the print file to stop until the operator acknowledges the message by entering a START_BATCH_DEVICE command on NOS/VE and a GO command on NOS. If no operator is controlling the I/O station, output of a print file terminates when a printer message is detected. The print file is held until the operator explicitly selects it to print.
PRINT	The line is printed using the *p* format effector.
DISCARD	The line is discarded.

Default is PRINT.

Responses

IO station <name> defined.

--ERROR-- DEFINE_USER_I_O_STATION definitions may not be included in a Terminal Definition Procedure addressed by a DEFINE_LINE command.

--ERROR-- Only one DEFUIOS defined I/O station may be defined in a Terminal Definition Procedure.

--ERROR-- DEFINE_I_O_STATION, DEFINE_USER_I_O_STATION or DEFINE_NP_BATCH_STATION commands may not be intermixed in the same Terminal Definition Procedure.

--FATAL-- Not enough memory is currently available for required table space.

Remarks

DEFINE_USER_I_O_STATION also generates a name for the I/O station by the concatenation of the following:

The string $IOSTATION_.

The last six hexadecimal digits of the DI's system ID.

A 4-digit decimal number in the range of 0000 through 9999. The DI software assigns this number consecutively for each $IOSTATION specification encountered. The first number assigned is 0000 (0000 follows 9999 thereafter).

For example, an I/O station connected to a DI system of ID 0800251FE029 that has last assigned number 1234 is named $IOSTATION_1FE029_1235.

Examples

The following command defines an operator-configured private I/O station that is controlled by control facility SCFS109 and is to print printer messages.

```
define_user_i_o_station control_facility=scfs109,..
p_m_action=print
```

Terminal User Procedure Commands

This section contains descriptions of commands that can be used in terminal user procedures (TUPs) only. Currently, there are two such commands:

SET_PAD_MESSAGE (SETPM)
PUT_STRING (PUTS)

The following commands (documented in the CDCNET Terminal Interface Usage manual) can be used in TUPs, and can be executed from a terminal.

ACTIVATE_X_PERSONAL_COMPUTER (ACTXPC)
CHANGE_CONNECTION_ATTRIBUTES (CHACA)
CHANGE_TERMINAL_ATTRIBUTES (CHATA)
CHANGE_WORKING_CONNECTION (CHAWC)
CREATE_CONNECTION (CREC)
DEFINE_PASSTHROUGH_TITLES (DEFPT)
DELETE_CONNECTION (DELC)
DISPLAY_COMMAND_INFORMATION (DISCI)
DISPLAY_COMMAND_LIST (DISCL)
DISPLAY_CONNECTIONS (DISC)
DISPLAY_CONNECTION_ATTRIBUTES (DISCA)
DISPLAY_SERVICES (DISS)
DISPLAY_TERMINAL_ATTRIBUTES (DISTA)
DO
REQUEST_NETWORK_OPERATOR (REQNO)

PUT_STRING (PUTS)

Purpose

This command is used within a terminal user procedure (TUP) to send a message either to the terminal or to the connected service. You cannot issue this command interactively from your terminal.

Format

PUT_STRING
  STRING = string
  DESTINATION = keyword value

Parameters

STRING (S)

Contains a message enclosed in single quotes. You can use as many as 80 characters in this message.

DESTINATION (D)

Identifies where you are sending the message. The following keyword values are allowed:

Keyword Value	Description
CONNECTION (C)	Sends the message to your service via the working connection.
TERMINAL (T)	Sends the message to your terminal.

If you do not select a destination, the network uses TERMINAL.

Remarks

- You cannot use the network command character in a PUT_STRING string.
- When putting a string to the service (D=C), the network treats the message like other data input from the terminal by forwarding it to the connected service.

  NOTE

  Within a TUP that creates a connection, do not try to put more than one string to the service after the CREATE_CONNECTION command. The second and subsequent PUT_STRING commands may not work.

  If the TUP is executed after you are connected to a service (it does not include a CREC command), you can send multiple PUT_STRING commands to the service.

- When putting a string to the terminal (D=T), the terminal displays the message in single-spaced format.

Examples

Site administrators sometimes create a terminal user procedure for a specific automatic login sequence. From a TUP, the following PUT_STRING command notifies the terminal user of what is happening:

```
put_string string='Logging into NOS/VE now. Please read ..
your mail for scheduling news.'
```

At the terminal, this string reads:

```
Logging into NOS/VE now. Please read your mail for scheduling news.
```

Then, another PUT_STRING command sends the login information to the service.

```
put_string string=',usernam,password,veiaf' destination=connection
```

The following PUT_STRING command contains a DEFINE_PASSTHROUGH_TITLES command as a string. The DEFINE_PASSTHROUGH_TITLES command is used to register a title for the passthrough connection in the Interactive Passthrough Gateway.

```
put_string string='define_passthrough_titles title=vepass' ..
  destination=connection
```

SET_PAD_MESSAGE (SETPM)

Purpose

Enables you to modify the CCITT and non-CCITT parameters of your public data network (PDN) (or packet assembler/disassembler (PAD) concentrators). Converts the parameter numbers and values into an X.29 set PAD message and sends it to the PAD. Parameter reference numbers and values are restricted to the range of 0 through 27. If non-CCITT parameters are included, they must follow CCITT parameters (when present), and the national marker must be included. This command may be executed only from a terminal user procedure.

Format

SET_PAD_MESSAGE
VALUE = list 1..63 of list 2 of integer

Parameters

VALUE (V)

A list of each PAD reference number followed by the value.

To effectively support CDCNET attributes, the X.25 Asynchronous TIP depends on the proper functioning on the X.29 PAD and the settings of the PAD parameter reference numbers. Table 8-1 shows PAD parameters and their default settings. Assuming no changes to the default terminal and connection attributes, the X.25 Asynchronous TIP attempts to set the following PAD parameter reference numbers at initial connection time. Use the SET_PAD_MESSAGE command only if you want to change these settings.

Remarks

The X.25 Asynchronous TIP treats *any* CCITT X.3 reference number modified by the SETPM command as an X.3 reference number that cannot be mapped. As a result, you should not use the SETPM command to set X.3 parameters that are mapped to VTP attributes, as results may be unpredictable.

Table 8-2 correlates PAD parameters to the corresponding CDCNET attributes. The X.25 Asynchronous TIP recomputes the PAD parameter values each time the CDCNET attributes are changed (by terminal user, application or when terminating transparent mode). If the computed values are different (previously computed values are maintained for each virtual circuit) a set PAD message is sent to the PDN PAD with the updated values.

Examples

The following command causes a set PAD message to be sent to the PAD that changes CCITT reference 3 (data forwarding signal) to a 2 (CR). CCITT reference 0 is CCITT-defined separator between Recommendation X.3 parameters and non-CCITT parameters. 33 is the Data Network Identification Code (DNIC) for TELENET. TELENET reference 63 (8-bit transparent) will be set to 0 (enabled).

```
SETPM,value=((3,2),(0,33),(63,0))
```

Table 8-1. Default PAD Parameter Settings

PAD Reference	Description	Default Setting/Remarks
1	PAD recall using a character	Decimal 1. Allows PAD recall using the DEL character.
2	Echo	Decimal 1. Causes the PAD to echo received characters to the start-stop mode DTE.
3	Selection of data forwarding signal	Decimal 34. Causes forwarding of data by the PAD upon entry of the ELC (default is CR) and the EPC (default is LF).
4	Selection of idle timer delay	Decimal 0. Indicates no data forwarding on timeout.
5	Ancillary device control	Decimal 0. Indicates no use of XON(DC1) and XOFF(DC3).
6	Control of PAD service signals	This reference number is never modified or referenced by the X.25 Asynchronous TIP.
7	Selection of operation of PAD on receipt of break signal from the start-stop mode DTE	Decimal 21 (1+4+16). Indicates that the PAD sends an interrupt packet to the packet mode DTE (1), sends an indication of break PAD message to the packet mode DTE (4), and discards output to the start-stop mode DTE (16) when a break signal is received.
8	Discard output	Decimal 0. Indicates normal data delivery to the start-stop mode DTE.
9	Padding after carriage return (CR)	Decimal 0. Indicates that the PAD will never perform padding after a carriage return.
10	Line folding	Decimal 0. Indicates that the PAD will never perform line folding.
11	Binary speed of start-stop mode DTE	This is a read-only parameter. It is never modified, and is referenced when computing FFD, CRD and LFD NULs.

(Continued)

Table 8-1. Default PAD Parameter Settings *(Continued)*

PAD Reference	Description	Default Setting/Remarks
12[1]	Flow control of the PAD by the start-stop mode DTE	Decimal 0. Indicates no use of XON(DC1) and XOFF(OFF).
13[1]	Line feed insertion after carriage return	Decimal 4. Indicates that the PAD inserts a line feed after echo of CR to start-stop mode DTE.
14[1]	Padding after line feed	Decimal 0. The PAD never performs padding after line feeds.
15[1]	Editing	Decimal 0. Indicates no use of editing in the data transfer state.
16[1]	Character delete	Never modified or referenced by the X.25 Asynchronous TIP.
17[1]	Line delete	Never modified or referenced by the X.25 Asynchronous TIP.
18[1]	Line display	Never modified or referenced by the X.25 Asynchronous TIP.
19[1]	Editing PAD service signals	Never modified or referenced by the X.25 Asynchronous TIP.
20[1]	Echo mask	Never modified or referenced by the X.25 Asynchronous TIP.
21[1]	Parity treatment	Never modified or referenced by the X.25 Asynchronous TIP.
22[1]	Page wait	Never modified or referenced by the X.25 Asynchronous TIP.

1. These PAD parameter reference numbers provide additional user facilities which are not necessarily provided in all PADs.

2. If the PAD returns an error PAD message in response to the setting of reference #13 (line feed insertion after carriage return), the X.25 Asynchronous TIP performs the cursor positioning itself. If an error PAD message is received in response to a setting of reference #3 (selection of data forwarding signal), the TIP sets reference #3 to 126. Any other errors reported by the PAD are ignored, since all other initial parameter settings are mandated by CCITT Recommendation X.3.

Table 8-2. Effect of CDCNET Terminal Attributes on PAD Settings

PAD Reference	CDCNET Attribute(s)	Effect on Setting
1	IEM	If the input editing mode (IEM) is TRANSPARENT, reference #1 (PAD recall using a character) is set to 0. Otherwise, reference #1 is set to 1.
2	E	If echoplex (E) is TRUE, reference #2 (echo) is set to 1. Otherwise, it is set to 0.
3	AC / IEM / ELC/ EPP / EPC / TTM / TLM / TCM / TFC / TTC	If the input editing mode is NORMAL, the setting for reference #3 is the aggregate forwarding signal determined by the attention character (AC), the end line character (ELC), and the end partial character (EPC), but only if end partial positioning (EPP) is selected. If the input editing mode is transparent, the setting for reference #3 is the aggregate forwarding signal based on the attention character (AC) and the type of transparent mode. If transparent timeout mode (TTM) is selected, reference #3 is set to 0, and reference #4 (selection of idle timer delay) to 8. If transparent length mode (TLM) is selected, reference #3 is set to 0, and reference #4 to 20. If transparent character mode (TCM) is selected and equal to forward (F), the transparent forwarding character(s) (TFC) are mapped to reference #3. If TCM is equal to terminate (T), the transparent terminating character(s) (TTC) are mapped to reference #3. If TCM is equal to forward terminate (FT), only the forwarding character (TFC) is mapped to reference #3. If no transparent mode is selected, reference #3 is set to 0 and reference #4 to 20. CDCNET defines the transparent forwarding and terminating characters (TFC/TTC) as 8-bit characters. Since CCITT has no provision for mapping 8-bit characters to reference #3, the X.25 Asynchronous TIP does not attempt to map these characters to reference #3 or reference #4 if the higher order bit is set.

(Continued)

Table 8-2. Effect of CDCNET Terminal Attributes on PAD Settings *(Continued)*

PAD Reference	CDCNET Attribute(s)	Effect on Setting
4	IEM	See discussion concerning the setting of reference #3. If the TIP cannot map a CDCNET character (AC/ELC/EPC/TTC/TFC) to reference #3, reference #4 is set to 20. If the computed value for reference #3 is rejected by the PAD (unsupported value), reference #4 is also set to 20.
5	CFC	If character flow control (CFC) is TRUE, reference #5 (ancillary device control) is set to 1; otherwise, it is set to 0.
12	CFC	If character flow control (CFC) is TRUE, reference #5 is set to 1; otherwise, it is set to 0.
13	IEM / E / ELC / ELP / EPC / EPP	If input editing mode (IEM) is NORMAL; echoplex (E) is TRUE; the end line character (ELC) is a carriage return (CR); end line positioning (ELP) is line feed (LF); the end partial character (EPC) is a carriage return (CR); and the end partial positioning (EPP) is line feed (LF), then reference #13 (line feed insertion after carriage return) is set to 4. Otherwise, reference #13 is set to 0.

Load Procedure Commands

This section contains descriptions of commands that can only be used in load procedures. Currently, there is one such command: DEFINE_VFU_LOAD_IMAGE.

DEFINE_VFU_LOAD_IMAGE (DEFVLI)

Purpose

Defines the format of Vertical Format Unit (VFU) load images. A series of these commands can be put into a procedure to define a single load image. This command can be specified only in a load procedure.

The procedure name can be referenced by the VFU_LOAD_PROCEDURE (VLP) parameter in the DEFINE_BATCH_DEVICE (DEFBD) and CHANGE_BATCH_DEVICE_ATTRIBUTES (CHABDA) commands, and as a file attribute of an output file. CDCNET software will concatenate the supplied procedure name with the string LOAD_PROCEDURE# and access the procedure as type LOAD_PROCEDURE. The procedure CDC_VFU is provided with the CDCNET software released by Control Data. CDC_VFU contains a load image suitable for each of the supported forms lengths of 8.5, 11, and 12 inches at print densities of both 6 and 8 lines-per-inch (suitable for printers with a terminal model value of C585V).

If you do not specify a default VFU load image for a printer that has a loadable VFU, the DI supporting the printer will use the Control Data-provided VFU load procedure (CDC_VFU). A VFU load procedure may be specified as a file attribute on an individual file by a user to override the default VFU load procedure.

Refer to the DEFINE_BATCH_DEVICE command description for information on when DEFINE_VFU_LOAD_IMAGE commands are processed.

Format

DEFINE_VFU_LOAD_IMAGE
LINE_NUMBER = list 1..50 of 1..255
CHANNEL = list 1..12 of 1..12

Parameters

LINE_NUMBER or LINE_NUMBERS (LN)

Specifies one or more lines of the paper form for which the channel numbers should be set.

CHANNEL or CHANNELS (C)

Specifies the channels which are set for the line.

Responses

VFU line/channel pair defined.

Remarks

The TIP software controlling a printer calculates page length as forms size times vertical print density.

You must specify top-of-form (channel 1) in the first line of the VFU data. Any lines not specified will be filled in by the command processor with zeroes, up to the page length specified by the file. If the bottom-of-form channel number and the auto page-eject channel number defined for the printer (see the DEFINE_PRINTER_MODEL_ATTRIBUTES command description) are not specified in the load procedure, the bottom-of-form channel will be provided at the line number with the value <page length minus 2>, and the auto page-eject channel will be provided at the line after that. If one of those channels is specified in the load procedure, the other will be defined so that the auto page-eject channel is in the line following the bottom-of-form channel.

If the operator or user changes the FORMS_SIZE or VERTICAL_PRINT_DENSITY batch device attributes (using the CHANGE_BATCH_DEVICE_ATTRIBUTES command), the bottom-of-form and auto page-eject channels will be moved to accommodate the change in number of lines on the form. If the auto page-eject option is changed, then the auto page-eject channel is removed or restored (depending on the option selected) in the VFU load image.

You may specify more lines in the VFU than the page length value, as determined by forms size times vertical print density. However, only channel information for page length number of lines will be sent to the printer's loadable VFU.

The top-of-form channel (channel 1), the bottom-of-form channel, and the auto page-eject channel may each be specified only once in the load procedure. The auto page-eject channel may not be defined for line number 1.

Examples

These examples of DEFINE_VFU_LOAD_IMAGE are from the default VFU load procedure, CDC_VFU. CDC_VFU defines printer control channels in the following lines:

Channel 1 in first line
Channel 2 in every 2 lines
Channel 3 in every 3 lines
Channel 4 in every 4 lines
Channel 5 in every 5 lines
Channel 6 in every 6 lines
Channel 7 in every 7 lines
Channel 9 in every 9 lines
Channel 10 in every 10 lines
Channel 11 in first line

```
define_vfu_load_image ..
  channel=1 ..
  line_number=1
```

```
define_vfu_load_image ..
  channel=2 ..
  line_number=(1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,..
               37,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,..
               71,73,75,77,79,81,83,85,87,89,91,93,95)

define_vfu_load_image ..
  channel=3 ..
  line_number=(1,4,7,10,13,16,19,22,25,28,31,34,37,40,43,46,49,..
               52,55,58,6164,67,70,73,76,79,82,85,88,91,94)
define_vfu_load_image ..
  channel=4 ..
  line_number=(1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,..
               61,65,69,73,77,81,85,89,93,)

define_vfu_load_image ..
  channel=5 ..
  line_number=(1,6,11,16,21,26,31,36,41,46,51,56,61,66,..
               71,76,86,91,96)

define_vfu_load_image ..
  channel=6 ..
  line_number=(1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91)

define_vfu_load_image ..
  channel=7 ..
  line_number=(1,8,15,22,29,36,43,50,57,64,71,78,85,92)

define_vfu_load_image ..
  channel=9 ..
  line_number=(1,10,19,28,37,46,55,64,73,82,91)

define_vfu_load_image ..
  channel=10 ..
  line_number=(1,11,21,31,41,51,61,71,81,91)

define_vfu_load_image ..
  channel=11 ..
  line_number=1
```

Utilities 9

Utilities 9

This chapter contains instructions, command descriptions, and messages generated by the following NOS utilities: Network Logfile Termination Utility (NLTERM) and Network Logfile List Utility (NLLIST). These utilities do not run under the Network Operator Utility (NETOU).

NLTERM is used on NOS to terminate CDCNET log files. NLLIST is used to generate a list of the terminated CDCNET log files.

NLTERM and NLLIST Command Descriptions

The following section provides descriptions of the commands used to invoke the NLTERM and NLLIST utilities. Following these descriptions are instructions for using the utilities.

NLTERM (Terminate CDCNET Log File)

Purpose Invokes NLTERM. Terminates CDCNET network log files, and renames the terminated log files. For instructions on using NLTERM once the utility is invoked, refer to Using the Network Log File Termination Utility in this chapter.

Format **NLTERM**
OP = keyword value
NM = file name
L = file name

Parameters *OP*

Specifies the environment in which NLTERM and its parameters will run. This parameter is needed only if you want to change the default option for your origin type. The following keyword values are allowed.

Keyword Value	Description
K	Parameters will be entered from host console.
T	Parameters will be entered from interactive terminal using full screen interface. If the T parameter is specified, the L parameter is used.
Z	Specifies that NLTERM is run with no interactive interface. In that case, NLTERM subcommands are specified only with the NLTERM command itself. In Z mode, you can only terminate the log file. You cannot perform other NLTERM activities, such as routing and purging files.

If you enter NLTERM at a host console, default is K. At an interactive terminal, default is T. For batch jobs, default is Z. Entering only NLTERM at a terminal enters you into a full screen NLTERM session. Refer to instructions on using NLTERM in this chapter.

If you want to use the interactive terminal interface, which uses screen formatting, make sure that your terminal is in screen mode rather than line mode. When you enter just NLTERM from a terminal without specifying the T parameter, you will receive the following message if the terminal is in line mode:

```
NLTERM - LINE MODE IS NOT SUPPORTED, USE SCREEN
```

If you do not have screen mode at your terminal, you have to enter all the parameters with the NLTERM command using the Z option. If you have type-ahead defined for your terminal, the cursor will go on to the next line or start over at the same line when you come to the end of a field.

NM

Specifies the name of a NOS permanent file that will receive the contents of the terminated network log file. Names may be from 1- to 5-alphanumeric characters long. NLTERM always attaches the prefix NL to file names. For example, the name LOG3 is named NLLOG3 by NLTERM.

If this parameter is not specified, a default file name is generated by NLTERM in the format

NLxmmdd

where:

x	The sequence number of the file. There are 36 possible values for x: A through Z, followed by 0 through 9 (see note).
mmdd	The month and day on which the network log file was terminated.

NOTE

NLTERM creates default file names, up to a limit of 36 file names per day. If more than 36 files are terminated in one day, you must supply a specific file name. NLTERM batch jobs attempting to terminate more than 36 files will not terminate the file, and will abort after sending an error message.

L

Specifies the name of the file to receive the list of terminated network log files. Parameters are used as initial values in both displays. L is a local file. This parameter is only meaningful if the NLTERM command list is entered through the host console or interactive terminal interface. The default file name is LIST.

Remarks

You can change the values of the NM and L parameters during an NLTERM session, if you want to change the name of the permanent file to receive a terminated log file, or the name of the local file to receive the listing of terminated log files.

Examples

```
nlterm,op=t,nm=file1.
```

NLLIST (List Terminated CDCNET Log Files)

Purpose Lists previously terminated network log files. The LIST subcommand in the NLTERM interactive or K display interface performs the same function as NLLIST.

Format **NLLIST**
L = file name

Parameters *L*

Name of file to receive listing of previously terminated network log files. Specified as a local file name. Default depends on environment in which NLLIST was entered. The default file name is LIST, unless NLLIST is submitted as a batch job (default is OUTPUT). After the NLLIST command executes, use the NOS command ROUTE to route the local file to a printer or you can copy the local file to your screen using the COPY command or NOS Full Screen Editor.

Remarks The LIST subcommand in the NLTERM interface performs the same function as NLLIST. Refer to instructions for using NLTERM.

Examples This example sends the list of terminated log files to a file called LOGFIL2.

```
nllist,l=logfil2.
```

Using the Network Log File Termination Utility (NLTERM) (NOS Only)

NLTERM terminates the currently active network log file and renames the terminated network log file as a NOS permanent file. NLTERM can be run as part of a daily system closedown process submitted as a batch job. The file name you specify can either be one you specify or a name generated by NLTERM. The NLTERM utility also provides a list of previously terminated network log files.

NLTERM has several subcommands.

GO	Terminates the currently active log file using the current network log file name parameter value. If this command fails after the name is changed, then the TERM command can be used (see TERM description).
LIST	Generates a list of previously terminated network log files. The list is then displayed on the screen and written to a file specified by the L (LIST) on the NLTERM command. Performs the same function as the NLLIST command.
PURGE	Purges the terminated network log file that is specified by the network log file name parameter value.
OUT	Routes the file specified by the L (LIST) parameter to the printer.
CLEAR	Returns the NLTERM command list to the screen, replacing the display of the list of terminated log files generated by the LIST command.
STOP	Terminates NLTERM processing normally.
TERM	Recovers a network log file if an error occurs while the log file is being terminated.

The NLTERM Utility may be invoked in three environments: from a host console, from an interactive terminal, or as a batch job. Once you access the NLTERM Utility, you may use the NLTERM subcommands. These subcommands may be entered either through the K display or interactive full screen interface.

NOTE

- In order for NLTERM and NLLIST to access the network log files, they must run under a user name validated by the NETOU MODVAL validation bit.
- A log file that is currently recording log messages can be partially reformatted for viewing using Network Performance Analyzer (NPA) commands. First, you reformat the log file using the REFORMAT_CDCNET_LOG_FILE (REFCLF) command, then use the CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) command to create reports on the log file. The network log file continues to record log messages, but you cannot directly observe this process. Any NPA reports you receive will only include information on the network log file up to the time it was reformatted, and do not include log messages added to the file after accessing NPA.
- You cannot invoke NLTERM during a NETOU session. To access NLTERM, you must exit from your current connection with NETOU.

1. Log in to IAF.
2. Enter the NLTERM command to invoke the NLTERM Utility. You can just enter NLTERM, or NLTERM with its optional parameters. Refer to the NLTERM command description for parameter descriptions.
3. Once NLTERM is accessed, you may enter NLTERM subcommands. The next subsections describe NLTERM's use from an interactive terminal and a host console. Messages that NLTERM generates are documented later in this chapter.

To run NLLIST, which has no K-display interface or full screen interface, enter the NLLIST command, as shown in the NLLIST command description in this chapter.

The function of NLLIST can be performed by the NLTERM subcommand LIST.

Using NLTERM at an Interactive Terminal

NLTERM uses screen formatting when run on an interactive terminal. Whether NLTERM can run under screen formatting at your site depends on the version of screen formatting that your site supports at terminals. NLTERM should run on any screen format-supported terminal that has at least a 23-line screen.

The main NLTERM interactive terminal display is shown in figure 9-1. Most of this display remains on the screen while you use NLTERM. For all terminal displays in this section, areas displayed in inverse video are marked by brackets ([ ]).

```
                 [ *** NETWORK LOG FILE TERMINATION *** ]
    OPTIONS                  DESCRIPTION
     [ NM ]= _______         Name of permanent file. Overrides automatic
                             naming (1-5 characters).
     [ L  ]= _______         File to receive output (1-7 characters).
     ______________________________________________________________

                      [ *** NLTERM COMMAND LIST *** ]
    [ GO     ] Perform log file termination processing using the
    [        ] current NM option value.
    [ STOP   ] Terminate utility processing normally.
    [ LIST   ] List the previously terminated log files on the
    [        ] the screen and on the file specified by the L option.
    [ CLEAR  ] Return the Command list to the screen.
    [ OUT    ] Route the file specified by the L option to the
    [        ] printer.
    [ PURGE  ] Purge the file specified by the NM option from the
    [        ] catalog.
    _______________________________________________________________

     Command - _________      Press [NEXT] to execute command.
     __________________________________________________ {Response Line}
```

Figure 9-1. Main NLTERM Interactive Display

Input is accepted through three places in the terminal display: the NM field, the L field, and the Command line. The display shows the current values for the NM and L parameters and allows you to change the values.

To change the NM or L value, position the cursor in the blank next to NM or L and enter the file name. The NM and L commands can also be used to change the values. When you start using NLTERM, the cursor is on the Command line. Enter NLTERM subcommands in the blank area next to Command.

The response line displays command responses and error messages. For example, if you try to provide a permanent file name that is seven characters long for the NM value, you will receive the following message on the response line.

```
LOG FILE NAME MUST BE 1 THROUGH 5 CHARACTERS IN LENGTH
```

The area where the command list is displayed also displays the list of previously terminated log files. The command list initially appears in this area. When you enter the LIST command, the command list is replaced by the list of terminated network log files. An example of a list of terminated, renamed log files displayed after entering a LIST command is shown in figure 9-2.

```
[ NO.]     [PFNAME ]   [  DATE   ] [  TIME   ]   [ LENGTH]

  1.        NLA0331     84/03/31    08.03.32        1073
  2.        NLB0331     84/03/31    12.02.24         376
  3.        NLA0401     84/04/01    08.01.40        1572
                           :
```

Figure 9-2. List of Terminated Log Files

The list of terminated log files contains the following information.

Field	Description
NO.	Order in which the logfiles are listed.
PFNAME	Name of the network log file. You may access the file as you would a permanent file.
DATE	Date network log file was terminated.
TIME	Time network log file was terminated.
LENGTH	Length of file, measured in PRUs.

NLTERM at K Display

The screen displays and command entry process differ slightly from those used at an interactive terminal. The K display NLTERM interface uses both left and right screens of the host console display. All K display screen formats are compatible with a 721 terminal being used as a host console as well as a CC545 console. The left screen is displayed at all times, and shows the last command executed, any messages from the utility, and current values for the NM and L parameters. The right screen shows two different displays: the NLTERM command list, and the list of terminated network log files, displayed after you enter a LIST command.

The format of the left screen NLTERM display is shown in figure 9-3.

```
                    *** NETWORK LOG FILE TERMINATION ***

OPTIONS                    DESCRIPTION

NM =                       NAME OF PERMANENT FILE.  OVERRIDES AUTOMATIC
                           NAMING (1-5 CHARACTERS).

L  = LIST                  FILE TO RECEIVE OUTPUT (1-7 CHARACTERS).

COMMAND - NM=NEWFILE    {command entry line}
LOG FILE NAME MUST BE 1-5 CHARACTERS IN LENGTH {response line}
```

Figure 9-3. Main NLTERM Display at K Display

NM specifies name of terminated log file. L specifies the name of the file to receive a listing of terminated log files. This example shows a command being entered on the command entry line. The user tried to assign a file name of NEWFILE to the file to receive the terminated log file output. NLTERM rejected this file name because the name was too long, since permanent file names provided for NLTERM can only be 1- through 5-characters long. NLTERM alerts the user to this error by displaying a message on the response line.

The initial right screen display (prior to entry of commands) contains the NLTERM command list. You can refer to this list when entering NLTERM commands. This list of commands is replaced by the list of previously terminated log files generated when the LIST command is executed. The list of terminated log files is arranged the same as that for the NLTERM display at an interactive terminal. Refer to that subsection for field descriptions for the list of terminated log files.

Terminating a Log File

To terminate a log file at an interactive terminal or K display:

1. Provide the name of the permanent file that will receive the terminated log file in the blank next to NM at a terminal, or by entering the command NM=file_name at the K display. You may also use the name automatically generated by NLTERM. The permanent file name may be 1- through 5-characters long. The prefix NL will be attached to the beginning of every file name.

2. Enter the GO command on the command entry line.

Listing Terminated Log Files

To get a list of terminated log files:

1. Check the current value of the L parameter (the file designated to receive the list output). If desired, change the value by entering the new local file name in the blank next to the L at a terminal, or by entering the command L=file name at the K display. The file name may be from 1- through 7-characters long, and must be a local file.

2. Enter the LIST command on the command entry line.

Printing a List of Terminated Log Files

To print a list of terminated log files:

1. Perform the LIST command as directed above.

2. Enter the OUT command on the command entry line. The list file will be routed to a printer.

Purging Terminated Log Files

To purge a terminated log file:

1. Provide the name of the permanent file to be purged in the blank next to NM at a terminal, or by entering the command NM=file_name at the K display. The permanent file name may be 1- through 5-characters long. The prefix NL will be attached to the beginning of every file name.

2. Enter the PURGE command on the command entry line.

Recovery from Incomplete Termination (the TERM Command)

There are two steps to network log file termination.

1. Changing the name of the network log file to the one provided by the user or to the name generated by the NLTERM Utility.

2. Writing a termination record to the file. This termination record contains information about the file's termination, such as the current time and date and the machine identification of the mainframe on which NLTERM runs.

Errors in processing may occur at either of these two steps. When errors occur before or during changing of the file name, no special processing is necessary. But if errors occur after the name change but before or while the termination record is written, special processing is necessary to properly terminate the file.

The TERM command recovers a network log file that has been terminated incompletely. The TERM command writes the termination record to a file whose name has already been changed. The error condition that prevented the original termination of this file must be corrected before the TERM command is used.

To use the TERM command:

1. Identify the error that occurred when you attempted to terminate the log file. Any error responses you receive on the response line should point to the problem.

2. Correct the error condition.

3. Specify the name of the file to which the termination record is to be written in the blank next to NM, or specify NM=file_name on the command line.

4. Enter the TERM command on the NLTERM command entry line.

NLTERM Utility Messages

This subsection contains the messages generated by the NLTERM Utility. The following messages will appear as a command response for the host console and terminal interfaces. Any errors causing termination of the utility will also appear in the user and system dayfile. If the Z option has been specified on the NLTERM command, these messages will all appear in the dayfile.

o IS AN ILLEGAL OR DUPLICATE OPTION

Description: The option o is illegal or has already been specified.

Action: Correct or remove the option from the command and rerun the job.

p IS AN ILLEGAL OR DUPLICATE PARAMETER

Description: The parameter p is illegal or has already been specified.

Action: Correct or remove the parameter from the command and rerun the job.

p PARAMETER MUST BE FOLLOWED BY AN EQUAL SIGN

Description: The parameter p must be followed by an equal sign.

Action: Correct the parameter specification and rerun the job.

A PARAMETER VALUE MUST BE SPECIFIED

Description: A parameter value must be specified with the L or NM parameter.

Action: Change the NM or L parameter so that a value is specified, and rerun the job.

LOG FILE NAME nm CONTAINS AN ILLEGAL CHARACTER

Description: The log file name nm contains a nonalphanumeric.

Action: Change the file name so that it contains only alphanumeric characters.

LOG FILE NAME MUST BE 5 CHARACTERS OR LESS

Description: The log file name specified by the NM parameter must be 1- to 5-characters in length.

Action: Change the log file name so that it is 1- to 5-characters in length and rerun the job.

ONE OPTION MUST BE SPECIFIED WHEN OP IS SPECIFIED

Description: A parameter value must be specified with the OP parameter.

Action: Change the OP parameter so that one parameter value is specified, and rerun the job.

TOO MANY OPTIONS ARE SPECIFIED

Description: Only one option can be specified.

Action: Correct the options parameter so that only one option is specified and rerun the job.

AN EQUAL SIGN MUST FOLLOW PARAMETER

Description: The NM and L parameters require an equal sign to follow the parameter.

Action: Change the parameter so that an equal sign follows it and reenter it.

AN ILLEGAL COMMAND IS SPECIFIED

Description: The command specified does not match any of the legal commands.

Action: Change the command so that it is one of the legal commands and reenter it.

CONTROL STATEMENT ERROR

Description: Indicates that a control statement error has occurred.

Action: Correct the control statement problem and rerun the job.

ILLEGAL CHARACTER FOUND IN COMMAND

Description: The command must be composed of alphabetic characters.

Action: Correct the command and reenter it.

ILLEGAL CHARACTER FOUND IN FILE NAME

Description: The file name specified must be composed of alphanumeric characters.

Action: Correct the file name and reenter it.

LOG FILE NAME MUST BE 1 – 5 CHARACTERS IN LENGTH

Description: The log file name specified must be 1- to 5-characters in length.

Action: Change the log file name so that it is 1- to 5-characters in length and reenter it.

OUTPUT FILE NAME MUST BE 1 – 7 CHARACTERS IN LENGTH

Description: The output file name must be 1- to 7-characters in length.

Action: Change the output file name so that it is 1- to 7-characters in length and reenter it.

A LOG FILE NAME MUST BE SPECIFIED ON A PURGE

Description: The PURGE subcommand requires that a file name specified.

Action: Set a file name with the NM parameter, or, using the full screen display, position the cursor to the NM field and enter a name. Then reenter the PURGE subcommand.

A LOG FILE NAME MUST BE SPECIFIED ON A TERM

Description: The TERM command requires that a file name is specified.

Action: Set a file name with the NM command or, using the full screen display, position the cursor to the NM field and enter a name. Then reenter the TERM subcommand.

CLEAR COMPLETE

Description: Indicates that the NLTERM subcommand list has been returned to the screen display.

Action: None.

LIST WRITTEN TO OUTPUT FILE nm

Description: Indicates that the LIST subcommand is complete and the list has been successfully written to the list file nm.

Action: None.

LOG FILE nm HAS BEEN TERMINATED

Description: Indicates that the GO subcommand has completed the termination of the file nm.

Action: None.

LOG FILE nm IS PURGED FROM THE CATALOG

Description: Indicates the completion of the PURGE subcommand.

Action: None.

LOG FILE NAME SET TO nm

Description: The NM parameter has set the file name to be used by the GO or TERM subcommands to the value nm.

Action: None.

OUTPUT FILE nm ROUTED TO THE PRINTER

Description: Indicates the completion of the OUT subcommand.

Action: None.

OUTPUT FILE NAME SET TO nm

Description: The L parameter has set the file name to be used by the LIST subcommand to the value nm.

Action: None.

ABNORMAL TERMINATION

Description: The utility has terminated abnormally.

Action: The reason for the abnormal termination is shown on the dayfile.

ATTEMPTING NETWORK NETON

Description: NLTERM is attempting to NETON to the network.

Action: None.

CIO ERROR ec DURING RETURN OF FILE nm

Description: CIO error ec occurred returning the file nm.

Action: Refer to volume 4 of the NOS 2 Reference Set for a description of the CIO error codes.

CIO ERROR ec DURING WRITE ON FILE nm

Description: CIO error ec occurred writing to file nm during log file termination. The log file name has been changed but the termination is not complete.

Action: Refer to volume 4 of the NOS 2 Reference Set for a description of the CIO error codes. After the error has been corrected, then the TERM subcommand can be used to complete the termination of the file.

CIO ERROR ec DURING WRITE TO OUTPUT FILE nm

Description: CIO error ec occurred writing to the output file nm.

Action: Refer to volume 4 of the NOS 2 Reference Set for a description of the CIO error codes.

CIO ERROR ec, EOI NOT FOUND ON FILE nm

Description: CIO error ec occurred on file nm indicating that no EOI exists on this file. This means that the file is a tape file or a problem exists with the disk file. The log file name is changed but the termination is not complete.

Action: If the file resides on tape move it to disk. Notify the site analyst if a problem exists with the disk file. Refer to volume 4 of the NOS 2 Reference Set for a description of the CIO error codes and of the SKIPEI macro. After the error is corrected, then the TERM subcommand can be used to complete the termination of the file.

ENDING NETWORK CONNECTION

Description: NLTERM is ending the network connection.

Action: None.

ERROR ec DURING ROUTE OF FILE nm

Description: Error ec occurred routing the output file nm to the printer.

Action: Refer to volume 4 of the NOS 2 Reference Set for the description of the ROUTE macro and its error codes.

ERROR ec OPENING PANEL pan

Description: Screen Formatting error ec occurred opening the panel pan.

Action: Refer to the Screen Formatting Reference manual for a description of the SFOPEN routine and its error codes. An error could occur if the PANELIB is not a local or system library, or if the panel is not on the PANELIB.

INTERNAL ERROR - rn

Description: The utility has detected an internal error condition in the routine rn.

Action: Follow the site defined procedures for reporting software problems.

LINE MODE IS NOT SUPPORTED, USE SCREEN MODE

Description: NLTERM is being run on a terminal that is not in screen mode or does not support screen mode.

Action: If the terminal model is one that is supported by the NOS SCREEN command, refer to the NOS 2 Reference Set volume 2, use it to make the terminal model known to the system. If the terminal is not supported by SCREEN, or it does not support screen mode, then use the K display for interactive processing, or set OP=Z and/or run NLTERM from a batch job for single-function processing.

LSF NOT AVAILABLE, NETWORK CONNECTION REJECTED

Description: The network has rejected the connection because the Log Server is not active. Log file termination continues.

Action: None.

NETWORK CONNECTION ACCEPTED

Description: NLTERM has accepted the connection from the network.

Action: None.

NETWORK CONNECTION ENDED

Description: The connection with the network is now ended.

Action: None.

NETWORK CONNECTION ENDED BY LSF

Description: The Network Log Server is in the process of returning the log file. The function of the connection with NLTERM is complete, therefore the connection is ended by the Network Log Server.

Action: None.

NETWORK CONNECTION INITIALIZED

Description: NLTERM has initialized the connection to the network.

Action: None.

NETWORK CONNECTION TO LSF TERMINATED PREMATURELY

Description: The connection being established by NLTERM to the Network Log Server was terminated before the connection was initialized. This could mean that the Network Log Server is no longer executing. Log file termination continues.

Action: None, unless the Network Log Server terminated abnormally. In that case, use the TERM subcommand.

NETWORK IDLE DOWN IN PROGRESS

Description: The network is in the process of shutting down. The network connection will end but log file termination continues.

Action: None.

NETWORK NETON ERROR, DUPLICATE NETON

Description: Two copies of NLTERM are executing at the same time both trying to terminate the currently active log file. NLTERM terminates abnormally.

Action: The new log file name that the Network Log Server created (as a result of the other copy of NLTERM terminating the log file) has been changed, and will need to be terminated using the TERM subcommand.

NETWORK NETON ERROR, APPLICATION DISABLED

Description: NLTERM has been disabled by the Network Operator. NLTERM terminates abnormally.

Action: The name of the currently active log file has been changed and will need to be terminated using the TERM subcommand.

NETWORK NETON UNSUCCESSFUL, NAM IS BUSY

Description: NLTERM is unable to NETON to the network to communicate to the Network Log Server to release the currently active log file that is to be terminated. Termination will still be attempted.

Action: None, unless termination of the file is unsuccessful. In that case, attempt to use the TERM subcommand to complete the termination of this file.

NETWORK NETON UNSUCCESSFUL, NAM IS UNAVAILABLE

Description: During log file termination processing NLTERM is unable to NETON to the network to communicate to the Network Log Server to release the currently active log file. Termination will still be attempted.

Action: None, unless termination of the file is unsuccessful. In that case, attempt to use the TERM subcommand to complete the termination of this file.

NETWORK NETON SUCCESSFUL

Description: NLTERM has successfully signed on to the network.

Action: None.

NETWORK PROTOCOL ERROR, REASON CODE IS rc

Description: A protocol error occurred, with reason code rc, during communication with network. NLTERM terminates abnormally.

Action: The name of the currently active log file has been changed and will need to be terminated using the TERM subcommand. Consult the site analyst regarding the protocol error.

NETWORK REQUESTED IMMEDIATE SHUTDOWN

Description: The network is shutting down immediately. NLTERM terminates abnormally.

Action: The name of the currently active log file has been changed and will need to be terminated using the TERM subcommand.

NO ACTIVITY ON NETWORK CONNECTION

Description: NLTERM has not received any supervisory messages in a length of time determined in NLTERM. NLTERM terminates abnormally.

Action: This may indicate an internal error in NLTERM. Consult the site analyst to see if other applications are communicating with the network or if something is wrong with the system.

NO LOG FILE EXISTS TO BE TERMINATED

Description: There is currently no active network log file to be terminated.

Action: None.

PFM ERROR ec ATTACHING FILE nm

Description: PFM error ec occurred trying to attach the file nm during log file termination.

Action: Refer to the applicable permanent file error diagnostic in volume 4 of the NOS 2 Reference Set. The name of the file has been changed but not terminated. The Network Log Server must release the file. Then use the TERM subcommand to complete the termination of this file.

PFM ERROR ec DURING CHANGE OF FILE nm

Description: PFM error ec occurred during the change of the currently active log file to file nm. The change occurs during log file termination.

Action: Refer to the applicable permanent file error diagnostic in volume 4 of the NOS 2 Reference Set.

PFM ERROR ec DURING INTERNAL CATLIST

Description: PFM error ec occurred during an internal catlist.

Action: Refer to the applicable permanent file error diagnostic in volume 4 of the NOS 2 Reference Set.

PFM ERROR ec DURING PURGE OF FILE nm

Description: PFM error ec occurred during the PURGE of file nm from the catalog.

Action: Refer to the applicable permanent file error diagnostic in volume 4 of the NOS 2 Reference Set.

TOO MANY TERMINATED LOG FILES EXIST, PROVIDE NAME

Description: Log file termination was attempted using a name generated by the NLTERM. NLTERM can generate only 36 names for each day. This number has been reached for this day.

Action: Assign a name or purge all of the log files which have been assigned for this day.

Appendixes

Glossary A

A

A-to-A

Refer to Application-to-Application.

Address Resolution Protocol (ARP)

A term used for reoutoing on a LAN. ARP is used to map IP addresses into Ethernet addresses. ARP is not required for connection to ARPANET or MILNET, but is useful in the LAN workstation environment.

Alarm

A log message that is routed to an operator. Any CDCNET log message may be designated as an alarm.

Alarm History

A chronological record of the alarms received at a network operator's alarm buffer since the start of an operations session. An alarm history may be displayed using a network operations command.

Application-to-Application

Can refer to either a type of link between two OSI layers, or a type of network processing:

1. An application-to-application link is an end-to-end link between an application layer of one system and the application layer of another for the exchange of information.

2. Application-to-application network processing that enables data to be exchanged between applications programs executing on different host computers or workstations.

ARP

Refer to Address Resolution Protocol.

ARPANET

A Defense Data Network (DDN) developed by the Defense Advanced Research Projects Agency. ARPANET supports research and development projects funded by the Department of Defense.

Asynchronous TIP

The terminal interface program (TIP) that configures terminal devices and establishes terminal attributes for a generic, asynchronous terminal connected to a device interface. The asynchronous TIP resides in a device interface that is configured to support asynchronous terminals.

Auto-Configured I/O Station

An I/O station that is logically configured and ready to use when the lines to which the devices in the I/O station become active and when a station operator connects to batch services. Contrast with Operator-configured I/O station. Configuring an auto-configured I/O station is possible when all the devices of the I/O station always connect to the same DI ports.

B

Batch Device

Individual devices in an I/O station controlled by batch services and protocols and used for batch input and/or output. Examples of batch devices include card readers, line printers, card punches and plotters.

Block

In the context of network communications, a portion or all of a message. A message is divided into blocks to facilitate buffering, transmission, error detection, and correction for variable-length datastreams. Differing block protocols apply to the host-to-device interface and the device-interface-to-terminal interfaces.

During input from a terminal, a block is a single transmission consisting of one or more lines of one or more messages.

During input to a service, a block is a single line consisting of part or all of a message. Terminal transmission blocks are divided into as many service input blocks as needed, until the message is completed.

During output from a host application program, a block is one or more lines. During output from a device interface to a terminal, a block is one terminal transmission buffer.

Board

Refer to Logic Board.

Break 2 Sequence

A series of interactive terminal keystrokes that cause interruption in the datastream, stopping delivery of a message or output from the host. Some terminals are equipped with a single key that causes a break 2 sequence. Refer to your terminal user's manual for your terminal's exact sequence.

Buffer

One of two structures for the storage of data in device interface memory. See also Data Buffer and Descriptor Buffer.

Byte

A group of contiguous bits. Unless prefixed (for example, a 6-bit byte), the term implies 8-bit groups. An 8-bit byte is sometimes called an octet. When used for encoding character data, a byte represents a single character.

C

Catenet

A group of connected CDCNET network solutions. This term is often used when referring to all the device interfaces and network solutions in a site's network.

Central Processor Unit (CPU)

The high-speed arithmetic processing unit that carries out the basic instructions required in program execution.

Channel

The physical link or logical path between a Mainframe Device Interface (MDI) and the network host computer, or between an Integrated Communication Adapter (ICA) and the Integrated Controller Interface (ICI) in the network host computer.

Client Application

Any network application that is authorized to initiate a connection to a server application. See also Server Application.

Clock Synchronization

A function that ensures that all device interfaces in a catenet are synchronized within 1 second of each other. Clock synchronization involves setting or resetting the master clock for the catenet (controlled by the Independent Clock ME) and synchronizing all of the device interface clocks in the catenet (controlled by the Dependent Clock ME in each device interface) according to the master clock. The DEFINE_SYSTEM command defines whether or not a device interface contains the Independent Clock ME.

On NOS, the device interface that contains the Independent Clock ME contains the master clock for the catenet, which synchronizes the rest of the clocks in the network.

On NOS/VE the Independent Clock ME is configured on the host.

Cluster Address

A sequence of bits, characters, or group of characters that identifies the location of a device (controller) that handles the remote communication processing for multiple (usually dumb) terminals or workstations.

Coaxial Cable

A transmission cable that provides large bandwidth and high data/low error rates. This cable contains a central carrier wire surrounded by fine copper mesh and/or an aluminum sleeve.

Command File

A NOS file of network operations commands. Commands in the command file can be executed using the EXECUTE_COMMAND_FILE. Similar to a procedure file.

Communication Line

A terminal line that establishes a complete communication circuit between a terminal or workstation and a CDCNET device interface.

Configuration

The process by which various computer-related resources are coordinated to function together. Under CDCNET, various types of configuration activities are performed.

1. Network configuration, whereby hosts, terminals, workstations, and unit record devices are interconnected into a network using CDCNET device interfaces and appropriate communications media.

2. Device interface hardware configurations, whereby decisions are made regarding which logic boards to install in a particular CDCNET device interface.

3. Device interface software configuration, whereby CYBER hosts decide which CDCNET software to down-line load into a specific CDCNET device interface.

4. Creation of device interface configuration files, whereby network administrators or communications consultants identify/describe the specific CDCNET device interfaces that reside in their networks and place this information in host-maintained permanent files.

See also Logical Configuration.

Configuration Command

A command that establishes, cancels, or redefines the configuration of a network component in the network's logical definition.

Configuration File

Refer to Configuration Procedure.

Configuration Procedure

A procedure containing configuration commands that configure the software in a device interface. Each device interface has a unique configuration file, which is read whenever the device interface is reset and loaded. Also known as configuration file.

Configure

To define the variable attributes of a CDCNET device (such as the device interface, a single board, network solution, communication line or gateway). Examples of configurable attributes include buffer sizes, line speeds, and logical names.

Congested

One of the operational states of a network solution or communication line; indicates excessive traffic. See also Congestion.

Congestion

A condition in which there is more message traffic on a network solution or communication line than the line's carrying capacity. Continued congestion results in lengthy message delay and discarding of new messages.

Control Facility

A NOS/VE service that monitors I/O stations and their batch devices, executes device and file control commands for the I/O station, and controls selection of files for output devices for the I/O station.

Cost

A relative measure assigned to a path (such as a network solution) that is used for transmitting data through a CDCNET-type network. The cost of each possible path is computed and stored into tables by the Routing Management Entity (ME). From these tables, the Routing ME determines the path that has the least cost. The path with the least cost is used to transmit data. The cost of a path may change depending upon the amount of congestion on the path. A congested network solution has a higher cost than an uncongested network solution.

Coupler

A hardware module on a Mainframe Device Interface (MDI) that connects a host's peripheral processor to CDCNET.

Coupler Node

A logical identification assigned to the coupler that connects a host channel and an MDI.

CPU

Refer to Central Processor Unit.

D

Data Buffer

A structure for storing user data in device interface memory. A pointer is associated with the first character of data in the buffer. Data buffer length is configurable. Contrast with Descriptor Buffer.

Datagram

A self-contained package of data carrying enough information to be routed from source to destination without reliance on earlier exchanges between source or destination and the transporting network.

DDN

Refer to Defense Data Network.

Deadman Timeout (DMTO)

A device interface hardware reset that occurs automatically if software does not work normally for 10 seconds.

Dedicated Line

A communication line that permanently connects a terminal to a device interface. Contrast with Switched Line.

Default

A preselected value supplied for a missing parameter upon the entry of a command or subcommand.

Defense Data Network (DDN)

A packet switching network provided by the Department of Defense (DOD) to meet its current and projected data communication requirements. It is based upon the Defense Advanced Research Projects Agency Network (ARPANET), an existing operational network.

Descriptor Buffer

A data structure used for chaining data buffers. Contrast with Data Buffer.

Diagnostic

1. Software and/or microcode that isolates failing hardware/software components within a CDCNET device interface.

2. A message indicating a malfunction within a CDCNET device interface or one of its related communications media.

Dial-Up Line

A communications circuit created by dialing a destination over a common carrier's switched lines.

Disabled

Cannot be used for normal network operation. Applies to boards, communication lines and network solutions.

DOD

Department of Defense.

Down

A status of suspended service.

Dump

Refer to Memory Dump.

Dump Analyzer

CDCNET troubleshooting software that enables communications support analysts to review detailed memory dumps generated by malfunctioning CDCNET device interfaces.

E

Echoplex

A procedure in which the receiving station automatically retransmits each character received so that the sender may verify the correctness of his transmission. This process usually occurs on asynchronous full-duplex communication lines; however, not all terminals on full-duplex communication lines are capable of echoplex operation.

EGP

Refer to Exterior Gateway Protocol.

Ethernet

A baseband local area network protocol developed by the Xerox Corporation. CDCNET supports an Ethernet-compatible network.

Exterior Gateway Protocol (EGP)

A TCP/IP protocol that allows for transfer and negotiation of routing information.

F

File Transfer Protocol (FTP)

TCP/IP protocol that provides the file transfer server and user functions.

Forms Code

A 1- through 6-character identifier associating a print file with a certain printer form which ensures output will be routed to a printer that prints in the format needed. For example, one printer at a site can be defined as using an 8 1/2 by 11-inch print form by specifying a forms code of DOC (document) on the command that configures the printer (DEFINE_BATCH_DEVICE). Another printer can be defined to print perforated checks and have a forms code of CHECKS, and one defined to print on carbon paper could have a forms code of CARBON. When output is routed to printers, the appropriate forms code (DOC, CHECKS, or CARBON) can be specified so that output will be printed by the appropriate printer.

FTP

Refer to File Transfer Protocol.

G

Gateway

A software interface between systems with different architectures and protocols.

Gateway Title

The logical title assigned to a gateway during logical configuration.

H

Hop

Within a network of interconnected gateways, a hop is the process of forwarding a packet from one gateway to another.

Host

Refer to Host Computer.

Host Computer

A mainframe computer system, connected to a communications network, that provides primary services such as database access, user application execution, or program compilation. For CDCNET, a host computer provides network support functions, including maintenance of device interface load files. Also called a host.

Host Console

The keyboard and display screen used to manage the host computer. Also used in CDCNET to access the Network Operator Utility (NETOU) to monitor and control the CDCNET. See also System Console.

Host Operating System

The host containing applications and maintenance software available to the device interface.

Host Service Name

A logical name for the host computer. The host service name is the name that terminal users provide when connecting to the host using the CREATE_CONNECTION command.

Host System

A mainframe computer and its operating system that provides applications and services to the computer network. CDCNET must have at least one host running NOS or dual-state NOS and NOS/VE.

Houston Automatic Spooling Program (HASP)

A job control protocol for transmitting data processing files and jobs between certain models of computers.

I

I/O Station

A logical grouping of batch devices into a single named unit for routing jobs and files to the batch devices and for controlling the devices. Devices belonging to an I/O station may all connect to the same line, to several lines on one device interface, or to lines distributed among several device interfaces.

ICA

Refer to Integrated Communications Adapter.

Independent Log Management Entity (Independent Log ME)

1. Also known as the recorder logging function. Software resident in a host-connected device interface that works with the Independent File Access ME to write log messages generated by network device interfaces to a file on a host called the network log file.

2. A service on NOS/VE host computers that writes log messages generated by network device interface to a host-resident file called the network log file.

Integrated Communications Adapter (ICA)

A hardware device that interconnects a single 16-bit Integrated Controller Interface (ICI) channel of a host computer with CDCNET. The ICA is installed in the CYBER 930 host computer mainframe.

Internet Protocol (IP)

A term used in DDN networks that refers to a connectionless, point-to-point protocol corresponding to the CDCNET Internet layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

IP

Refer to Internet Protocol.

Isolation

Identification of a failing hardware or software component.

K

K Display

A NOS host console display that enables operators to interact with various operating system utilities (for example, those controlling user validation and NAM subsystem interaction).

L

Line

A circuit that connects a terminal to a device interface. A line is dedicated to carrying data to and from that terminal. It does not carry data that is routed through the rest of the network, nor does it use the CDNA protocol. Also known as a communication line.

Log File

A file that is created and maintained by the operating system for storing error information and usage data concerning network elements.

Log Group

A logging function that is distributed among several device interfaces. A collection of device interfaces and the set of log messages associated with these device interfaces.

Log Management Entity (Log ME)

Software that manages the transmission and recording of log messages generated by device interface software. Consists of Dependent and Independent Log Management Entities. Dependent Log Management Entities, residing in device interfaces, are sources of log messages. Independent Log Management Entities, residing in a host-connected device interface, work with host applications or a NOS/VE host to write the log messages to the network's log file on the host.

Log Support Application (LSA)

Also known as the Dependent Log Management Entity and/or source logging function. Software that manages the generation and transmission of log messages generated by device interface software. Resident in every device interface.

Logging

The process of issuing messages for network activity and recording the messages in a log file.

Logic Board

A printed circuit board with data storage and/or processing components installed; sometimes called a board, card, or module.

Logical Configuration

The process of assigning names and values and setting variables throughout the CDCNET to define network elements (mainframes, terminals, lines, network solutions, device interfaces, gateways, and other elements), so that all network elements follow a uniform naming and addressing scheme. After logical configuration, network elements accept all data and commands directed to or through themselves, and reject all other data and commands. Also known as network definition.

Logical Name

A name assigned to a CDCNET component (device interface, network solution, communication line, gateway) in the logical defintion of the network. Many network operations commands refer to CDCNET components by their logical names.

Loopback Test

A failure management test that checks the integrity of a hardware element by sending data through the element and back again.

M

Mainframe Channel Interface (MCI)

An optional logic board within a CDCNET device interface that connects the device interface to a 12-bit CYBER host channel.

Mainframe Device Interface (MDI)

The standard CDCNET device interface variant that interconnects a 12-bit channel of host computers operating under NOS or NOS/VE with an Ethernet local area network.

Mainframe/Terminal Device Interface (MTI)

The standard CDCNET device interface variant that interconnects NOS and NOS/VE host computers with terminals, workstations, and unit record equipment without requiring a local area network.

MANAGE_CDCNET_CONFIGURATION (MANCC) Utility

A CDCNET host utility that helps create, edit, and display CDCNET configuration files.

Management Entities (MEs)

CDCNET software components that provide management functions for device interfaces such as control of errors and transmission of log messages.

Management Entity (ME)

CDCNET software that performs network management functions. CDCNET supports various MEs to perform specific network tasks.

MCI

Refer to Mainframe Channel Interface.

MDI

Refer to Mainframe Device Interface.

Memory Dump

The process and result of writing device interface memory to a host-resident file. Memory dumps are forced when the contents of device interface memory are at risk of being lost.

Metrics

Statistics which are collected and reported for CDCNET hardware and software components.

MILNET

A Defense Data Network (DDN) evolved from ARPANET that supports operational communication requirements.

MTI

Refer to Mainframe/Terminal Device Interface.

N

NAM

Refer to Network Access Method.

NAM K Display

A display on the host console screen that allows operator interface to Network Access Method (NAM). A CDCNET operator at the host console communicates with the CDCNET through the NAM K display.

NAM/VE

Refer to Network Access Method/Virtual Environment.

NDI

Refer to Network Device Interface.

NETOPS

A NOS user name that is used to store files used for CDCNET installation and CDCNET-host operations. NETOPS contains files created and written by NAM while NAM is operating, the network directory file (NETDIR), and the NAMSTRT procedure.

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 Architecture

A set of functional layers in which each layer performs a specific set of functions and services; together, the layers interact to provide total, end-to-end network operation. Each layer uses a protocol and has its relationship with other layers defined.

Network Definition

The process of assigning logical names to network components and assigning values to variable parameters for CDCNET software. See also Logical Configuration.

Network Device Interface (NDI)

The standard CDCNET device interface variant that transfers data between networks (for example, between two local area networks, between a local area network and a communications line, or between a local area network and a public data network).

Network Identifier

A unique identifier assigned to a network solution.

Network Log File

A file on a host computer that contains CDCNET log messages sent from the network's device interfaces and serves as a record of the network's activity.

Network Log Server (NETLS)

A CDCNET host application that writes CDCNET log messages generated by device interfaces to the network log file on the host.

Network Logfile Termination (NLTERM) Utility

A CDCNET host utility on NOS that terminates the currently-active network log file to which NETLS is writing log messages, and renames the terminated log file. NLTERM also provides information about previously-terminated log files as an aid in managing log files.

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 by sending commands to the network's device interfaces. A network operator may perform these tasks from a host console or a remote terminal.

Network Operator Utility (NETOU)

A group of programs residing on a host computer and in a (NOS) mainframe device interface or mainframe terminal interface connected to the mainframe. NETOU allows a network operator to access, monitor, control and configure a CDCNET from the host console or a remote terminal. Using NETOU, network operators can send CDCNET operations commands to specific device interfaces or to all the device interfaces in the network.

Network Performance Analyzer (NPA)

The CDCNET software utility that generates statistical reports based on its analysis of the network log file or generates event/error reports based on log messages in the network log file.

Network Products Gateway

A gateway that allows information transfer between CDCNET and a non-CDNA host such as a NOS host. File transfers between NOS hosts over CDCNET require Network Products gateways to be defined in the MDIs connected to the hosts.

Network Products (NP)

Programs that run under NOS in a host mainframe to allow data and computer applications to be transmitted from the mainframe through a computer network. Network Products include Network Access Method (NAM) and Network Definition Language (NDL). Network Products and CDCNET have different architectures. For hosts to send data through CDCNET, the Mainframe Device Interfaces connected to the mainframes must have gateways to translate between Network Products and CDCNET protocols.

Network Products Terminal Gateway

A gateway that allows both interactive and remote batch terminal users to connect to a NOS host through CDCNET (by specifying the appropriate service title on the CREATE_CONNECTION command). There are two parts to the NP Terminal gateway: the Interactive Virtual Terminal gateway (IVT gateway) and the Remote Batch Facility gateway (RBF gateway). The batch gateway is dependent on the interactive gateway. If a network configuration is going to support terminal connections to NOS, the MDI or MTI connected to the NOS host must contain an NP Terminal gateway.

Network Solution

A communications medium over which data is transmitted between interconnected network resources, and which uses CDCNET protocols. A network solution differs from other communications lines because it is shared by multiple network resources (it is not solely dedicated to the handling of data transmissions between a single pair of network resources). Network solutions differ from trunks because they can carry network management traffic such as log and alarm messages.

NLTERM

Refer to Network Logfile Termination Utility (NLTERM).

NP

Refer to Network Products.

NP IVT Gateway

Network Products Interactive Virtual Terminal Gateway. A program which runs in a Mainframe Device Interface (MDI) or Mainframe Terminal Device Interface (MTI) connected to a host mainframe, and which allows the host mainframe to send applications through CDCNET to interactive terminals. The gateway acts as a protocol converter between the host's Network Products protocols and CDCNET protocols. Also known as the NP terminal gateway.

NP Terminal Gateway

Refer to NP IVT Gateway.

O

Octet

An 8-bit byte.

Online Diagnostics

Optional diagnostics for the device interface that can be executed while the device interface is connected to and operating as part of the CDCNET.

Online Loader

A CDCNET service that loads software into device interfaces when the software is needed while the network is operational, as opposed to initial loader, which loads software into device interfaces only when they are started up (initialized).

Operations Station

The remote terminal or host console from which CDCNET network operations are performed through the Network Operations Utility (NETOU).

Operator-Configured I/O Station

An I/O station that is logically configured when an I/O station operator invokes a terminal definition procedure (TDP) to define the I/O station. The station operator must define the I/O station before it can be used, and the devices in the I/O station are not active until the TDP executes. Contrast with Auto-configured I/O Station. Configuring an Operator-configured I/O station is necessary when the devices of an I/O station do not always connect to the same device interface port. An example of an Operator-configured I/O station is a dial-up HASP workstation.

Operator Console

An interactive terminal in an I/O station that can be used to control the other batch devices in the I/O. On NOS/VE, the operator console is used for entering OPERATE_STATION (OPES) utility commands to control the devices. On NOS, the operator console is used for entering Remote Batch Facility (RBF) commands to control the devices.

P

Physical Name

A name assigned to a hardware device in a device interface: boards, ports, and memory banks, such as $CIM3 (physical name for CIM board in slot 3) and $LIM5_PORT2 (physical name for second port on LIM board in slot 5.)

Physical Record Unit (PRU)

The amount of information transmitted by a single physical operation of a specified device. For mass storage files, a PRU is 64 central memory words (640 characters); for magnetic tape files, the size of the PRU depends upon the tape format. A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

Port

The physical connection on the device interface through which data is transferred to/from the device interface. Each port is numbered and supports a single communication line.

Primary MDI

The Mainframe Device Interface (MDI) to which the operator sends commands and receives responses and alarms. At any time, only one MDI can communicate with the operator.

Private I/O Station

An I/O station used to submit and receive jobs and output files only for the user that is operating it. A station operator must monitor and control the I/O station for it to be active. Contrast with Public I/O Station.

Private Memory Module (PMM)

The logic board within a CDCNET device interface that provides additional random access memory dedicated for use by the main processor board (MPB) of the device interface.

Programming System Report (PSR)

An official report to Control Data of a problem with Control Data software. A PSR can be sent to Control Data either in hard-copy form, or by using the on-line SOLVER program.

Protocol

A set of conventions that must be followed to achieve complete communications between the computer-related resources in a network. A protocol can reflect the following:

1. A set of predefined coding sequences, such as the control byte envelopes added to (or removed from) data exchanged with a terminal.

2. A set of data addressing and division methods, such as the block mechanism used between a network application program and Network Access Method.

3. A set of procedures that control communications, such as the supervisory message sequences used between a network application program and Network Access Method.

PRU

Refer to Physical Record Unit.

Public Data Network (PDN)

A commercial packet-switching network that supports the communications interface described in CCITT protocol X.25.

Public I/O Station

An I/O station shared by many users who may submit jobs through it and receive the output from these jobs at it. The operator who controls a public I/O station does not own the files sent to or read from it. Routing of output files for a public I/O station is controlled through the I/O station's name. A station operator does not have to monitor and control a public I/O station for it to be active. Contrast with Private I/O Station.

R

Radix

The base of a number system. For example, 2 is the binary system radix and 10 is the decimal system radix.

RBF

Refer to Remote Batch Facility.

Recorder Log Group

A logging function in which device interfaces that are sources of log messages report their log messages to a device interface which works with a host application to record the log messages in a network log file. The Independent Log ME controls the log recording function.

Remote Batch Facility (RBF)

The network applications software that supports remote batch processing (remote job entry) on NOS.

RS-232-C

An Electrical and Electronic Industries Association (EIA) standard that describes the interface between terminals or other Data Terminal Equipment (DTE) and modems or other Data Communications Equipment (DCE) employing a serial binary interchange.

RS-449

1. A physical interface standard for data communications used with high speeds and long communication lines.

2. A newer standard than RS-232-C, also used for serial communications. Eventually meant to replace RS-232-C, but backward compatibility is specified in RS-449.

S

SCL

Refer to System Command Language.

SCL Comment

A comment within a SCL command. The comment is enclosed by quotation marks and is ignored during command processing.

Server TELNET

Provides a mechanism for an interactive terminal on a foreign host to communicate with the interactive services of NOS/VE.

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. Services have names. Terminal users connecting to a host are connecting to a service. An example of a service is the Interactive Facility (IAF) on a host.

SOLVER

An online utility maintained by Control Data that contains a database of reported software problems and solutions. SOLVER can be used for writing a PSR to report a problem with software.

Source Log Group

A logging function in which device interfaces that are sources of log messages maintain a list of log messages which they will send to recorder device interfaces. The source logging function is controlled by the Dependent Log ME, also known as Log Support Application (LSA).

Station Operator

A person in charge of controlling batch devices in an I/O station by sending commands to the equipment from the station operator console. On NOS/VE, the station operator uses OPERATE_STATION (OPES) utility commands to control the devices. On NOS, the station operator uses the Remote Batch Facility (RBF) commands to control the devices.

Statistics

Refer to Metrics.

Status

Information about the current state of a network component: Device Interface (DI), the hardware components (boards, ports) of a device interface, lines and network solutions connected to the device interface, and device interface software.

Status Command

A command that requests and displays the operational status of a particular network component, such as a device interface or a network solution.

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.

Synchronous Command Entry Mode

A command control mechanism that prevents operators from entering a command before a previously sent command has executed and returned a response.

System Address

The unique address assigned to a device interface in the network. The system address corresponds to the system title, so that commands and data sent by system title are received at the proper device interface address. See also System Identifier.

System Command Language (SCL)

The NOS/VE command language on which CDCNET network operations, configuration and terminal user commands are based.

System Console

A component of a host operating system that is used to monitor and control the operating system. The system console can also be used to monitor and control CDCNET through the Network Operations Facility (NOF). See also Host Console.

System Identifier

At the time of its manufacture, each device interface is assigned a unique 48-bit identification number from a pool of numbers allocated to Control Data by Xerox. This number is written into battery-backed RAM and is used throughout the catenet as the system identifier for that device interface.

The system identifier is used as the Ethernet address for any system that is locally connected to one or more Ethernet network solutions.

See also System Address.

System Main Memory (SMM)

A device interface board with 1024K byte increments of dynamic RAM accessible by all interfaces and the resident main processor board (MPB).

System Title

The title assigned to a device interface during logical configuration. This title corresponds to the device interface's system address, so that commands sent to a device interface by system are received at the proper device interface address.

SYSTEMX

A username that is used to store files for NOS and CDCNET installation and operations.

T

TCP

Refer to Transmission Control Protocol.

TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is the name given to a suite of protocols that support the ARPANET community. TCP/IP protocol implementation is required within CDCNET for connectability to Defense Data Networks (MILNET or ARPANET) and to workstations that use TCP/IP.

TDI

Refer to Terminal Device Interface.

TDP

Refer to Terminal Definition Procedure.

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 Device Interface (TDI)

The standard CDCNET device interface variant that interconnects terminals, workstations, and unit record devices with an Ethernet local area network.

Terminal Interface Program (TIP)

CDCNET software that resides in terminal device interfaces (TDIs) and enables terminals/workstations that employ specific terminal protocols (such as asynch, HASP, and IBM 3270) to communicate in CDCNET networks.

Terminal-to-Application (T-to-A)

A type of network processing that enables the exchange of data between applications programs that reside on host computers and user terminals or workstations. In this case, protocol conversions occur so that transmitted data is understood both at the host and at the terminal or workstation.

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.

Test

Software and/or microcode that provides detection and confidence capabilities. Also known as a diagnostic.

TIP

Refer to Terminal Interface Program.

Title

A string of 1- through 256-ASCII characters that identifiy a network service component such as a device interface or a gateway. The Directory Management Entity refers to the component by its title.

A name used to identify services available in the network. Titles are known throughout the catenet. Contrast with logical names, which are local to individual device interfaces.

Transmission Control Protocol (TCP)

A term used in DDN networks that refers to an end-to-end, connection-oriented protocol corresponding to the CDCNET Transport layer. This protocol is required for connection to MILNET, ARPANET, and TCP/IP workstations.

Transmission Media

Provides the physical channel used to interconnect device interfaces in a network.

Trunk

A logical definition of a line and the communications software that allows the line to carry data between communications controllers. These controllers could be device interfaces or devices for other networks. Trunks going to other networks, such as DECNET or SNA, are not recognized as network solutions.

TUP

Refer to Terminal User Procedure.

U

ULP

Refer to Upper Layer Protocols.

Upper Layer Protocols (ULP)

A collective term for layers 5, 6, and 7 of the Open System Interconnection (OSI) network reference model.

User TELNET

Allows a CDCNET terminal to connect to a foreign host's interactive service.

V

Version

A four-digit hexadecimal number indicating the release version of the software loaded in a device interface.

Virtual Circuit

A connection between a source and a receiver in a network that may be realized by different circuit configurations during data transmission. Also called a logical circuit.

W

Well-Known Port

Ports used in TCP to name the ends of logical connections which carry long-term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. A contact port is sometimes referred to as a well-known port.

Wildcard Characters

Characters that can be used in place of other characters as variables. Wildcard characters can be used to replace single characters, to replace strings of characters, or to match characters to those specified in a list.

X

X.25 Gateway

A gateway used to transfer data from a host connected to CDCNET to a host in another network at the other end of the X.25 circuit. The X.25 gateway allows host-to-host (A-to-A) connections to take place over an X.25 cicuit. A-to-A connections over X.25 circuits are provided by the Network Products applications.

Xerox Networking System

Provides an efficient way of connecting devices to Ethernet LANs and internetworking through gateways.

XNS

Refer to Xerox Networking System.

Network Operations Command Types B

This appendix is intended as a guide to CDCNET Network Operations command types, to help you understand the relationship between the command names and their functions. Some commands are in several command types. For example, the CANCEL_SOURCE_ALARM_MESSAGE command can be used to cancel the current configuration of an alarm message reporting set-up, so it is in the cancel configuration command type. This command is also used for alarm control, and is found in the Alarm Control Command group as well. For the complete command and parameter descriptions, refer to chapters 6, 7, and 8 of this manual.

Session Control Commands

These commands help you do the following:

- Define and display parameters of your command and alarm environments for operations sessions.
- Define and display parameters common to one or more systems in the catenet.

NOS/VE Session Control Commands

ACTIVATE_ALARMS (ACTA)

DEACTIVATE_ALARMS (DEAA)

QUIT (QUI)

SEND_COMMAND (SENC)

All other NOS/VE commands (refer to SCL for NOS/VE System Interface manual)

NOS Session Control Commands

ACTIVATE_ALARMS (ACTA)

BYE

CHANGE_ALARM_ENVIRONMENT (CHAAE)

DEACTIVATE_ALARMS (DEAA)

DISPLAY_ALARM_ENVIRONMENT (DISAE)

DISPLAY_ALARM_HISTORY (DISAH)

DISPLAY_CATENET_TITLES (DISCT)

DISPLAY_CONNECTED_MDI (DISCM)

EXECUTE_COMMAND_FILE (EXECF)

GOODBYE

HELLO

INCLUDE_FILE (INCF)

LOGOUT

LOGIN

QUIT

RESTORE_ALARM_ENVIRONMENT (RESAE)

ROUTE_ALARM (ROUA)

ROUTE_COMMAND_RESPONSE (ROUCR)

SEND_COMMAND (SENC)

SEND_COMMAND_SEQUENCE (SENCS)

SET_COMMAND_MDI (SETCM)

Display Commands

General Display Commands

These commands provide general information about DI components that you may need to use throughout your operations sessions.

DISPLAY_DATE_AND_TIME (DISDAT)

DISPLAY_LOGICAL_NAMES (DISLN)

Display Status Commands

Display Status Commands display the operational status of the hardware devices, communication lines, network solutions, and communication software configured for a DI system.

DISPLAY_DIRECTORY_STATUS (DISDS)

DISPLAY_DI_SYSTEM_STATUS (DISDSS)

DISPLAY_HARDWARE_STATUS (DISHS)

DISPLAY_LINE_STATUS (DISLS)

DISPLAY_NETWORK_STATUS (DISNS)

DISPLAY_PASSTHROUGH_STATUS (DISPS)

DISPLAY_ROUTING_STATUS (DISRS)

DISPLAY_SOFTWARE_LOAD_STATUS (DISSLS)

DISPLAY_XNS_TRANSPORT_STATUS (DISXTS)

Display Configuration Commands

Each DI is configured through load media and DEFINE commands used in configuration files. The display configuration commands display the current values of DI configuration parameters defined through the load process and the DEFINE commands. Descriptions of the DEFINE commands are included in the Network Commands chapter of this manual.

The display configuration response names each configuration option by the same name as that used in the associated network definition command. Using this display, you can observe current configurations, and decide if you should change existing configurations.

DISPLAY_FILE_SUPPORT (DISFS)

DISPLAY_HDLC_NET_OPTIONS (DISHSO)

DISPLAY_HDLC_TRUNK_OPTIONS (DISHTO)

DISPLAY_LINE_OPTIONS (DISLO)

DISPLAY_NP_GW_OUTCALL_OPTIONS(DISNGOO)

DISPLAY_RECORDER_LOG_GROUP (DISRLG)

DISPLAY_SOURCE_ALARMS (DISSA)

DISPLAY_SOURCE_LOG_GROUP (DISSLG)

DISPLAY_SYSTEM_OPTIONS (DISSO)

DISPLAY_X25_GW_OUTCALL_OPTIONS (DISXOO)

DISPLAY_X25_NET_OPTIONS (DISXNO)

Network Control Commands

Clock Management Commands

These commands control time clocks for the entire network and for individual DIs.

SET_DATE_AND_TIME (SETDAT)

SYNCHRONIZE_CLOCK (SYNC)

Operator Message Control Commands

WRITE_TERMINAL_MESSAGE (WRITM)

Communication Control Commands

Communication control commands start or stop communications on communication trunks, networks and asynchronous lines. These commands address trunks, networks and lines by their logical names (see Network Operations concepts in chapter 1) by network definition commands.

START_LINE (STAL)

START_NETWORK (STAN)

START_NP_INTERFACE (STANI)

START_X25_ASYNCTIP (STAIA)

START_X25_GW (STAXG)

START_X25_INTERFACE (STAXI)

STOP_NP_GW

STOP_LINE (STOL)

STOP_NP_INTERFACE (STONI)

STOP_NETWORK (STON)

STOP_X25_ASYNCTIP (STOXA)

STOP_X25_GW (STOXG)

STOP_X25_INTERFACE (STOXI)

Statistics Control Commands

Statistics control commands start the collection, start the reporting, stop the collection, or stop the reporting of statistics for trunks, network solutions, communication lines, and communication software. You may choose which type of statistics you want: summary, expanded, debug, or all types.

If statistics are already started for lines, trunks, network solutions, or software, they are immediately reported, and the report is restarted.

START_LINE_METRICS (STALM)

START_NETWORK_METRICS (STANM)

START_PROCESS_METRICS (STAPM)

STOP_LINE_METRICS (STOLM)

STOP_NETWORK_METRICS (STONM)

STOP_PROCESS_METRICS (STOPM)

NPA Commands and Procedures Used in Obtaining Statistics

REFCLF (REFORMAT_CDCNET_LOG_FILE) Command

CRECAR (CREATE_CDCNET_ANALYSIS_REPORT) Procedure

Logging Control Commands

CANCEL_SOURCE_LOG_GROUP (CANSLG)

CHANGE_SOURCE_LOG_GROUP (CHASLG)

DEFINE_SOURCE_LOG_GROUP (DEFSLG)

DISPLAY_SOURCE_LOG_GROUP (DISSLG)

CANCEL_RECORDER_LOG_GROUP (CANRLG) (NOS Only)

DEFINE_RECORDER_LOG_GROUP (DEFRLG) (NOS Only)

DISPLAY_RECORDER_LOG_GROUP (DISRLG) (NOS Only)

Alarm Control Commands

ACTIVATE_ALARMS (ACTA)

CANCEL_SOURCE_ALARM_MESSAGE (CANSAM)

DEACTIVATE_ALARMS (DEAA)

DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM)

CHANGE_ALARM_ENVIRONMENT (CHAAE) (NOS Only)

DISPLAY_ALARM_ENVIRONMENT (DISAE) (NOS Only)

DISPLAY_ALARM_HISTORY (DISAH) (NOS Only)

RESTORE_ALARM_ENVIRONMENT (RESAE) (NOS Only)

ROUTE_ALARM (ROUA) (NOS Only)

Diagnostic Control Commands

CHANGE_ELEMENT_STATE (CHAES)

DISPLAY_TEST_STATUS (DISTS)

START_CIM_TEST (STACT)

START_ESCI_TEST (STAET)

START_LIM_TEST (STALT)

START_MCI_INLINE_TEST (STAMIT)

START_PORT_TEST (STAPT)

START_URI_TEST (STAUT)

STOP_CIM_TEST (STOCT)

STOP_ESCI_TEST (STOET)

STOP_LIM_TEST (STOLT)

STOP_MCI_INLINE_TEST (STOMIT)

STOP_PORT_TEST (STOPT)

STOP_URI_TEST (STOUT)

Configuration Commands

This section lists the define, cancel, and change configuration commands.

Define Configuration Commands

The define configuration commands are usually executed as part of a DI's configuration files. Most of these commands may also be executed on an operating network.

In most cases, to change the configuration of a network component, you must first use one of the cancel configuration commands to cancel the component's configuration, and then redefine the component's configuration using one of the network definition commands listed below. In some cases, you can directly change the configuration using a change configuration command, rather than first having to cancel the current configuration.

Exception List Commands

The following commands can be used only within an exception list for a CDCNET-type network. Exception list commands are used to define the load process and dumping conditions for CDCNET systems. There are two commands used for exception lists: one to specify the load process parameters for specific systems and one to specify the load process parameters for systems that do not have an explicit specification of their own. These commands are described in the CDCNET Configuration and Site Administration Guide.

DEFINE_BOOT_DEFAULTS (DEFBD)

DEFINE_EXCEPTION_SYSTEM (DEFES)

DI Commands Common to NOS/VE and NOS

The following commands can be used both in system configuration files and entered through NETOU to define and redefine network communication media, management entities, and interface software. The commands are used in CDCNET environments that have either a NOS/VE or a NOS host.

DEFINE_ETHER_NET (DEFEN)

DEFINE_ETHER_TRUNK (DEFET)

DEFINE_LINE (DEFL)

DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM)

DEFINE_SOURCE_LOG_GROUP (DEFSLG)

DEFINE_SYSTEM (DEFS)

DEFINE_TIP (DEFT)

DEFINE_X25_GW (DEFXG)

DEFINE_X25_INTERFACE (DEFXI)

DEFINE_X25_NET (DEFXN)

DEFINE_X25_TRUNK (DEFXT)

DI Commands (NOS Only)

The following commands are used only in CDCNET environments that interface to a NOS host. The commands can be used both in system configuration files or entered through NETOU to define and redefine the network management entities and Network Products (NP) interface software required to interface a NOS host and CDCNET.

DEFINE_FILE_SUPPORT (DEFFS)

DEFINE_NP_GW (DEFNG)

DEFINE_NP_INTERFACE (DEFNI)

DEFINE_NP_TERMINAL_GW (DEFNTG)

DEFINE_OPERATOR_SUPPORT (DEFNTG)

DEFINE_RECORDER_LOG_GROUP (DEFRLG)

Terminal Definition Procedure-Only Commands

The following commands are used to define interactive terminals and batch I/O stations. These commands can only be executed within terminal definition procedures (TDPs). They cannot be entered directly as commands using NETOU. One command, DEFINE_NP_BATCH_STATION, is used only for I/O stations that access NOS hosts. For more information on TDPs and their use in configuring terminals and I/O stations, refer to the CDCNET Configuration and Site Administration Guide.

DEFINE_BATCH_DEVICE (DEFBD)

DEFINE_I_O_STATION (DEFIOS)

DEFINE_NP_BATCH_STATION (DEFNBS) (NOS Only)

DEFINE_TERMINAL_DEVICE (DEFTD)

DEFINE_USER_I_O_STATION (DEFUIOS)

Cancel Configuration Commands

The following commands cancel the logical configuration of network communications media, network management entities, and interface software. Cancelling a logical configuration logically removes the component or function from the network. To use the trunk, network, line, or function again, you must redefine it using the DEFINE configuration commands described later in this chapter. To cancel the logical configuration of a trunk, network, or line, you must first deactivate the trunk, network, or line using the appropriate STOP command.

CANCEL_ETHER_NET (CANEN)

CANCEL_FILE_SUPPORT (CANFS) (NOS Only)

CANCEL_HDLC_TRUNK (CANHT)

CANCEL_LINE (CANL)

CANCEL_NP_INTERFACE (CANNI) (NOS Only)

CANCEL_OPERATOR_SUPPORT (CANOS) (NOS Only)

CANCEL_PASSTHROUGH_SERVICE (CANPS)

CANCEL_RECORDER_LOG_GROUP (CANRLG) (NOS Only)

CANCEL_REMOTE_LOAD_SUPPORT (CANRLS)

CANCEL_SOURCE_LOG_GROUP (CANSLG)

CANCEL_SOURCE_ALARM_MESSAGE (CANSAM)

CANCEL_X25_ASYNCTIP (CANXA)

CANCEL_X25_GW (CANXG)

CANCEL_X25_INTERFACE (CANXI)

CANCEL_X25_NET (CANXN)

CANCEL_X25_TRUNK (CANXT)

Change Configuration Commands

The following commands make changes to parameters defining the network's logical configuration without your having to cancel the existing configuration and redefine it.

CHANGE_SOURCE_ALARM_MESSAGE (CHASAM)

CHANGE_SOURCE_LOG_GROUP (CHASLG)

CHANGE_SYSTEM (CHAS)

CHANGE_PASSTHROUGH_SERVICE (CHAPS)

DI State Control Commands

These commands control the operational state of the DI. Currently, one DI state control command is supported.

KILL_SYSTEM (KILS)

Software Loading and Unloading Commands

Software loading and unloading commands allow you to control the presence of CDCNET software in DIs.

LOAD_MODULE (LOAM)

UNLOAD_MODULE (UNLM)

Network Log File Management Commands and Utilities

The following commands and utilities are used to control the CDCNET network log file. Different commands are used on NOS/VE and NOS.

NOS/VE Network Log File Commands

On NOS/VE, the network log file is controlled by two commands in the NOS/VE START_UP_COMMANDS file.

ACTIVATE_NETWORK_LOG

DEACTIVATE_NETWORK_LOG

Refer to the Network Interface manual of the NOS/VE System Analyst Reference Set for more information on these commands.

NOS Network Log File Commands

On NOS, the network log file is controlled by the Network Logfile Termination Utility (NLTERM). NLTERM is invoked by the NLTERM command, control activities are performed through NLTERM subcommands. A separate utility, NLLIST, is used to generate a list of the terminated network log files. NLTERM and NLLIST cannot be invoked during an active NETOU session.

NLLIST

NLTERM

NLTERM Subcommands:

- GO
- LIST
- PURGE
- OUT
- CLEAR
- STOP
- TERM

Procedures to Enhance Operator Environment C

This section describes NOS/VE SCL procedures that can be used to enhance the NETOU environment. These procedures use the NETOU functions described in chapter 3. CDC does not currently support these procedures in its released software. You may write and install these procedures at your site. For more information on SCL Procedures, refer to the SCL for NOS/VE Language Definition manual.

DISPLAY_SYSTEM_NAMES (DISSN) Procedure

DISPLAY_SYSTEM_NAMES uses the $MATCHING_NAMES function to send a command to the set of systems matching a specified name. The procedure accepts names with wildcards, which allows you to send the command to all systems that match the wildcard name. For more information on wildcards, see the Wildcard Characters section of chapter 2.

DISPLAY_SYSTEM_NAMES has the following format:

```
PROC display_system_names, dissn ();

  x=$matching_names('[A-Z]*')
  for i=$variable(x,lower_bound) to $variable(x,upper_bound)
    display_value x(i)
  forend
PROCEND display_system_names
```

CREATE_COMMAND_CONNECTION (CRECC) Procedure

CREATE_COMMAND_CONNECTION sends one or more commands to CDCNET systems. A system name or a list of system names is specified when invoking this procedure. When the CRECC procedure is invoked, NETOU commands can be entered as command, rather than by enclosing commands as string values within SEND_COMMAND. Entering a slash (/) sends the command to the NOS/VE host rather than to a DI or DIs. The command QUIT ends the procedure. CREATE_COMMAND_CONNECTION has the following format.

```
PROC create_command_connection, crecc (system, s : list of name = $required)

  create_variable command_status k=status
  create_variable command k=string
  accept_line v=command i=$input p=$parameter(system)
  while $translate(lower_to_upper,$substr(command,1,3) <> 'QUI' do
    if $substr(command,1) - '/' then
      include_line command status=command_status
    else
      include_line 'send_command c=command s='//$parameter(system)//' ..
        status=command_status'
    ifend
    if not command_status.normal then
      display_value command_status
    ifend
    accept_line v=command i=$input p=$parameter(system
  whilend
PROCEND create_command_connection
```

Index

Index

D

E

O

T

U

W

X

Comments (continued from other side)

?lease fold on dotted line;
;eal edges with tape only.

FOLI

'OLD

FOLD

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
Technology & Publications Division
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-6198

CDCNET Network Operations

60461520 D

We value your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

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

What programming languages do you use? ______________

How do you like this manual? Check those questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Do the illustrations help you?

Comments? If applicable, note page and paragraph. Use other side if needed.

Would you like a reply? ☐ Yes ☐ No

From:

Name ______________ Company ______________

Address ______________ Date ______________

______________ Phone ______________

____

Please send program listing and output if applicable to your comment.

Network Operations Commands Index

CONTROL DATA