CONTROL DATA

60000414

CDCNET Commands

CDCNET

Commands

Reference

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 60000414

Manual History

Revision	System Version	PSR Level	Date
A	1.5.1	739	December 1989
B	1.5.2	750	June 1990
C	1.5.3	765	January 1991
D	1.6.1	780	September 1991
E	1.7.1	803	October 1992
F	1.8.1	826	December 1993
G	1.8.3	857	March 1996
H	1.8.4	871	December 1997

This revision includes changes for NOS/VE version 1.8.4 at PSR level 871.

Changed pages are as follows:

Title Page/2, 15 through 18, 8-123/8-124, Chapter 13 Divider Page, 13-35, Quick Index-7

Revision letters I, O, Q, S, X, and Z are not used.

DISPLAY_TRACE_PROCESS
DISPLAY_X25_INCALL
LOAD_FILE
START_TRACE_PROCESS
UNLOAD_FILE

Contents

Figures

Tables

About This Manual

This manual contains the formats, descriptions, and examples for the following commands:

- Network operations
- Network Performance Analyzer (NPA)
- Dump Analyzer

Audience

This manual is written for the person performing Control Data Distributed Communications Network (CDCNET) activities, such as starting and stopping communication lines and displaying operational status of network components. Communication support analysts may use some of the commands described within during troubleshooting. Customer engineers may also use this manual for reference.

The reader should have knowledge of CONTROL DATA® NOS/VE (Network Operating System/Virtual Environment) and/or NOS (Network Operating System) 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 documents CDCNET commands.

- Chapter 1 contains an introduction of the manual contents.
- Chapter 2 contains the network operations session control commands.
- Chapter 3 contains the network operations ADD commands.
- Chapter 4 contains the network operations CANCEL commands.
- Chapter 5 contains the network operations CHANGE commands.
- Chapter 6 contains the network operations DEFINE commands.
- Chapter 7 contains the network operations DELETE commands.
- Chapter 8 contains the network operations DISPLAY commands.
- Chapter 9 contains the network operations EXEET, HELP, IDLSD, IDLTS, and KIL commands.
- Chapter 10 contains the network operations LOAD and RESUME commands.
- Chapter 11 contains the network operations SENDD and SETDT commands.
- Chapter 12 contains the network operations START commands.
- Chapter 13 contains the network operations STOP commands.
- Chapter 14 contains the network operations SYNC, UNLF, UNLM, and WRITM commands.
- Chapter 15 contains the terminal definition/user procedure commands.
- Chapter 16 contains the load procedure commands.
- Chapter 17 contains the configuration file management commands.
- Chapter 18 contains the NPA commands.
- Chapter 19 contains the dump analyzer commands.

The appendixes contain additional information about CDCNET commands.

- Appendix A contains a glossary of terms.
- Appendix B contains the CDCNET (NLTERM, NLLIST, and NLPURGE) utility commands.
- Appendix C contains the CDCNET file and procedure management (NETFM, NETBDF, NETMDF, and NETRDF) utilities.
- Appendix D contains the Manage CDCNET Configuration (MANCC) utility commands.
- Appendix E contains the Network Configuration Utility (NETCU).
- Appendix F contains the character set.

- Appendix G contains the DI reset codes.

Conventions

The following conventions are used in this manual:

- The terms CDCNET and network are used interchangeably.
- Commands, parameters, or descriptive text that apply to only NOS or NOS/VE operating systems are marked either NOS only or NOS/VE only.
- The NOS operating system recognizes only the abbreviated form of command names and parameters.
- In general, all NPA commands on NOS/VE follow the standard System Command Language (SCL). See the NOS/VE Commands and Functions manual.
- All NPA commands on NOS are CYBER Control Language (CCL) commands (except REFCLF) and follow the CCL syntax conventions (see the NOS Version 2.3 Screen Formatting Reference manual).
- On NOS/VE, a blank (space) may be used in place of a comma (,).
- All numbers are decimal unless otherwise specified.
- A vertical bar in the margin indicates a technical change.
- The following typographical conventions are used:

boldface	Command and subcommand names are shown in boldface type when illustrating a format. Required parameters are also in boldface.
italics	Optional parameters in a command or subcommand format are shown in italics.
UPPERCASE	Uppercase indicates names of commands, subcommands, and parameters (and their abbreviations). Names of variables, files, and system constants are shown in uppercase when they occur in text.
`example`	Examples, including user input, are printed in a typeface that simulates computer output.

Terminal Use

The terminal you use allows you to perform some procedures interactively in either screen mode or line mode, or both.

If you have a video display terminal (VDT), you can choose to perform some procedures in either screen or line mode. If you do not have a VDT, you are limited to performing the procedures in line mode.

Screen mode operation (NOS only) provides you with a full screen menu display of required and optional parameters. All you need to do is fill in the appropriate values on the corresponding lines of the menu display.

Line mode operation allows you to enter the command and all the parameters at one time.

Special Keyboard Entries

CDCNET supports a variety of terminals. Although the individual keys used in our examples may not be present on your keyboard, your terminal contains keys that provide similar functions.

See the CDCNET Terminal Interface manual for more information.

Related CDCNET Manuals

Manual Abstracts

Following is a brief description of each CDCNET manual.

Conceptual Overview	Discusses CDCNET in conceptual terms. It provides a broad view of CDCNET that explains the theoretical nature of this product. It does not attempt to define which particular product capabilities and features are currently available and which ones will follow in subsequent releases.
Product Descriptions	Provides reference, planning, and training information for customers who own or are interested in owning CDCNET products, and for Control Data personnel who use or work on CDCNET. The manual describes hardware and software products, provides information on how to select and use various types of network cables, and provides network configuration examples.
Terminal Interface	This is the primary manual for end-users who use interactive terminals to access computer services connected to CDCNET. The manual explains general terminal interface concepts, terminal commands and attributes, and connection attributes. For the advanced user, site administrator, and network analysts it also covers more advanced topics such as virtual and transparent modes, resolving communications problems, and the various terminal protocols supported by CDCNET.
Access Guide	This online manual guides the novice user through the process of accessing and using computer services through CDCNET. It includes procedures for connecting, disconnecting, and managing connections; displaying and changing terminal attributes; and terminal user exception processing. The more experienced user can find additional related information in the CDCNET Terminal Interface manual.
TCP/IP Programming Interfaces and Applications	Describes how to access the utilities that implement the TCP/IP protocols through CDCNET. The manual assumes the user is familiar with CDCNET terminal and connection attributes; knows the service title to access; and has some working knowledge and understanding of TCP/IP protocols.
Batch Device User Guide	Describes how to operate batch devices connected to CDCNET. It assumes the user is familiar with NOS and/or NOS/VE operating systems and with CDCNET access to these operating systems. The manual defines the concepts of I/O stations and provides the procedures for defining and controlling these stations. The online manual is available with NOS/VE and NOS operating systems.

Hardware Installation and Troubleshooting	Contains hardware installation procedures and troubleshooting guidelines for CDCNET hardware products and associated I/O cables. The manual is intended for individuals who install and check out CDCNET hardware products, operate them, add options to them, and maintain them.
Configuration Guide	Documents how to configure CDCNET software after it is installed on an operating system, and describes the responsibilities of the CDCNET network administrator. This manual also documents the Manage CDCNET Configuration Utility (MANCC), a utility for creating and editing files defining a CDCNET network.
DI Dump Analyzer	This manual is an online version of the DI Dump Analyzer section of the CDCNET Network Operations and Analysis manual. The manual is for CDCNET analysts who are familiar with Control Data host computer operating system concepts and operations. The manual describes how to use information from the Analyze CDCNET Dump (ANACD) utility to help troubleshoot network problems. Available with NOS/VE only.
Network Operations and Analysis	This manual documents how to monitor, control, and reconfigure CDCNET using the CDCNET Network Operator Utility (NETOU). The Network Operations section walks an operator through operations concepts, basic and advanced operations activities, and elementary troubleshooting decisions. The Network Analysis section describes the tools and methods used to analyze CDCNET performance including: instructions for using the CDCNET DI Dump Analyzer, a list of DI reset codes, a map of fixed address memory, and definitions of important system data structures. The NPA section of the manual provides information on how to generate various types of NPA reports and provides examples and descriptions of all NPA reports.

Diagnostic Messages	This manual is for network operators, network analysts, and programmers. The manual provides sorted lists of diagnostic messages and command responses issued by the CDCNET software. The primary sorted list of diagnostic messages describes the event causing each message and the appropriate user action. The primary sorted list of command responses describes the event causing the command response. Secondary sorted lists of diagnostic messages and command responses provide a cross reference of diagnostic message number and command response number to the CDCNET software products that issue the messages or command responses. The printed version of this manual is no longer available. However, a copy of the messages file can be printed on site. Available with both NOS/VE and NOS operating systems.
Commands	This manual contains all of the CDCNET Operator/Analyst commands. This manual is intended for operators, systems analysts, support engineers, and other experienced users.
CDCNET Network Management Station	This manual documents how to install, configure, and operate the CDCNET Network Management Station. The manual is for CDCNET operators and aministrators having previous experience as a UNIX system administrator.

Manual History

Not all sites find it convenient or expedient to install each new version and PSR level of CDCNET software. This presents a problem in maintaining sets of manuals that reflect installed software when later versions of CDCNET software are available but not installed. The following CDCNET Manual History table helps users to assemble and maintain the appropriate documentation by indicating which manual revisions support each release of CDCNET.

Manual/Audience Matrix

The CDCNET Manual/Audience matrix helps site planners, administrators, and users to determine their CDCNET documentation needs. The matrix categorizes each manual according to its type: overview, reference, tutorial, and so on. It then defines the audience of each manual in general terms: customer, end-user, LAN installer, and so on. Sites may have different audience designations for their audience, or may combine user functions.

CDCNET MANUAL HISTORY RELEASE 1.5.1 — 1.8.1	CDCNET RELEASE DATE/VERSION/PSR LEVEL					
	DEC. '89 R1.5.1 L739	JUN. '90 R1.5.2 L750	JAN. '91 R1.5.3 L765	SEP. '91 R1.6.1 L780	OCT. '92 R1.7.1 L803	DEC. '93 R1.8.1 L826
CDCNET MANUALS	MANUAL REVISION					
CONCEPTUAL OVERVIEW 60461540		C				
PRODUCT DESCRIPTIONS 60460590	D		E			
TERMINAL INTERFACE 60463850	F	G		H		J
ACCESS GUIDE (ONLINE NOS) CDCNETA	X					
ACCESS GUIDE (ONLINE NOS/VE) CDCNET_ACCESS	X	X				
TCP/IP APPLICATIONS 60000214	E	F	G	H	J	K
BATCH DEVICE USER GUIDE 60463863	F		G			
BATCH DEVICE USER GUIDE (ONLINE NOS/VE) CDCNET_BATCH	X		X			
BATCH DEVICE USER GUIDE (ONLINE NOS) CDCNETB	X					
HARDWARE INSTALLATION AND TROUBLESHOOTING 60000348	B					
CONFIGURATION GUIDE 60461550	G	H	J	K	M	N
DI DUMP ANALYZER (ONLINE NOS/VE) ANACD	X					
NETWORK OPERATIONS AND ANALYSIS 60461520	G	H	J			
DIAGNOSTICS MESSAGES (ONLINE NOS) CNETMSG	X	X	X	X	X	X
DIAGNOSTICS MESSAGES (ONLINE NOS/VE) CDCNET_MSGS	X	X	X	X	X	X
CDCNET COMMANDS 60000414	A	B	C	D	E	F
CDCNET COMMANDS (ONLINE NOS/VE) CDCNET_COMMANDS				X	X	X
CDCNET NETWORK MANAGEMENT STATION 60000568			A	C	D	

NOTES:

M06026

MANUAL NOT AFFECTED BY THE RELEASE.

X INDICATES ONLINE MANUAL WAS UPDATED FOR THAT RELEASE.

SHADED BOXES INDICATE THE LATEST REVISION LEVEL FOR THE MANUAL.

CDCNET MANUAL/AUDIENCE MATRIX

CDCNET MANUALS	MANUAL TYPE	AUDIENCE: Customer	End-User	LAN Installer	Customer Engineer	Network Operator	CE Support Engineer	Network Analyst	Site Administrator	Programmer
Conceptual Overview	Overview	■			■		■	■	■	■
Product Descriptions	Reference	■			■		■	■	■	■
Terminal Interface	User Guide		■			■	■	■		■
Access Guide	User Guide		■				■	■		
TCP/IP Programming Interface and Applications	Reference		■		■		■	■	■	■
Batch Device User Guide	User Guide		■				■	■		
Hardware Installation And Troubleshooting	Maintenance	■		■	■	■	■	■		
Configuration Guide	Ref./Tutorial					■	■	■	■	
DI Dump Analyzer	Ref./Tutorial				■	■	■	■		
Commands	Reference				■	■	■	■	■	■
Network Operations And Analysis	Ref./Tutorial					■	■	■	■	
Diagnostics Messages	Reference				■	■	■	■	■	■
CDCNET Network Management Station	Ref./Tutorial				■	■	■	■	■	

M05353

Referenced Manuals

The following manuals are referenced in this manual and may contain helpful information.

Manual	Publication Number
Concurrent Maintenance Library for Virtual Environment	60000019
Network Access Method (NAM) Network Definition Language (NDL)	60480000
NOS Screen Formatting	60460430
NOS/VE System Usage	60464014
NOS/VE Commands and Functions	60464018
NOS Version 2 Analysis Handbook	60459300
NOS/VE Network Management	60463916

Ordering Manuals

Control Data manuals are available through Control Data Sales Offices or from:

Control Data
Literature and Distribution Services ARHLDS
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

You can also call (612)415-4478 or FAX your inquiry to (612)415-4359.

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, you can mail your comments to:

Control Data
Documentation Services ARH219
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

You can also submit your comments through SOLVER, an on-line 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 or not you would like a written response.

Customer Support Hotline

Control Data maintains a hotline to assist you in the use of our products. This hotline service is available to you for each product you license and cover with a Control Data software support agreement.

If you need help not provided by the documentation or find that a product does not perform as described, contact the hotline through one of the following methods:

Mail

Mailing Address	Shipping Address
Control Data ARH213 4201 Lexington Avenue N. St. Paul, MN 55126-6198	Control Data Central Software Support 4233 Lexington Avenue N. Arden Hills, MN 55126-6198

Telephone

Telephone Number	Countries
800-345-6628	USA and Canada (follow touchtone prompts)
612-415-4272	Other countries

Internet

Internet Address	Support Area
epix@css.cdc.com	EP/IX Operating System Support
irix@css.cdc.com	IRIX Operating System Support
eplx@css.cdc.com	EP/LX Operating System Support
nos@css.cdc.com	NOS Operating System Support
nosve@css.cdc.com	NOS/VE Operating System Support
networks@css.cdc.com	Network Support
compilers@css.cdc.com	Compiler Support
icem@css.cdc.com	ICEM Support
support@css.cdc.com	Miscellaneous
solveradm@css.cdc.com	SOLVER Validations
solver@css.cdc.com	Email interface to SOLVER

When you call or write the support hotline, please include the following information:

- Site Number (assigned for all hardware and software support requests)
- Operating system and level installed
- Machine type (for example, 4360-302, 910B-232)
- Product for which support is needed
- Details concerning the problem

This allows CSS to solve your problem as quickly and efficiently as possible.

SOLVER

As a Control Data support customer, you also receive a subscription to SOLVER, the online support service. SOLVER is an interactive tool with menus and help for all functions and prompts. After some initial interactive setup, a subset of SOLVER functions is also available via email. Email in a prescribed format can be sent to solver@css.cdc.com for automatic processing.

From SOLVER, you can report problems (PSRs) to Control Data, search with keywords to find similar problem reports from other sites, monitor progress against a PSR (including workarounds and fix availability), request software enhancements (RSEs), and write comments against the documentation. SOLVER also includes Installation Bulletins that document post-release highlights, warnings, and product update announcements.

For help with SOLVER or SOLVER validation information, contact CSS at the telephone numbers listed above or email to solveradm@css.cdc.com.

Introduction 1

Introduction 1

This chapter gives an overview of the commands used to configure and analyze a network. This includes the commands for:

- Network Operator Utility (NETOU)
  - Session Control
  - Network Operations
- Terminal Definition Procedure (TDP)
- Terminal User Procedure (TUP)
- Load Procedure
- Configuration File Management
- Network Configuration Utility (NETCU)
- Network Performance Analyzer (NPA)
- DI Dump Analyzer

Network Operator Utility (NETOU) Commands

The Network Operator Utility (NETOU) allows you to access CDCNET and run network operations commands. These commands let you inspect and significantly alter the network. Use these commands to:

- Monitor the status of network components.
- Change the operational state of network components.
- Manage network component failures.
- Configure network components logically.

Operations commands can be divided into the following types:

- Session control commands which are used to 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.
- Network commands which are used to monitor, control, and dynamically change the logical definition of network equipment.

You access NETOU to perform network operations activities from a remote terminal or from a 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.

Starting a NETOU Session

NOS/VE and NOS differ when it comes to starting a NETOU session. Assuming you are validated to use NETOU, you must first log onto the network host (NOS/VE or NOS) on which you have validation.

Under NOS/VE, NETOU is a system command you can enter in response to the NOS/VE system prompt. See the CDCNET Network Operations and Analysis manual for descriptions of this command's optional PROLOG and STATUS parameters.

Under NOS, specify NETOU for the application; if you are already logged onto the NOS IAF application, use one of the following commands to begin a NETOU session:

```
HELLO,NETOU
```

or

```
BYE,NETOU
```

or

```
APPSW,NETOU
```

A further distinction between using NETOU on NOS/VE and NOS is that under NOS/VE, the host runs its own versions of some CDCNET software. This software enables the host to manage a NETOU session between you and the network.

When using NETOU under NOS, you must establish a working relationship with a Mainframe Device Interface (MDI). This MDI, which you designate the Command MDI, coordinates your session with the other CDCNET systems on the network.

The SET_COMMAND_MDI Command (NOS only)

After you specify NETOU as the application, you see the NETOU banner and the prompt to enter the SET_COMMAND_MDI (SETCM) command. A list of the available MDIs is provided with the prompt.

The NETOU banner message and prompt are formatted as follows:

```
WELCOME TO NETWORK OPERATOR UTILITY
 CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1986, 1987, 1988.

                  STATUS OF CONNECTED MDIs
                  NODE    CURRENT    SYSTEM
                 NUMBER    STATE     TITLE

                   089   AVAILABLE   MDI_8B
                   088   AVAILABLE   MTI_F
                   093   AVAILABLE   MDI_A

More than one MDI available.
To select a DI, type:   SETCM MDI=MDINAME
MDINAME is optional, if not specified,
default is MDI_8B
```

If more than one MDI is available to act as Command MDI, you must enter the SETCM command to begin the NETOU session. A default Command MDI is assigned if you enter the SETCM command without the MDI parameter.

If there is only one MDI available to act as Command MDI, that MDI is selected for you by default; the banner and prompt are formatted as follows:

```
WELCOME TO NETWORK OPERATOR UTILITY
 CDCNET - COPYRIGHT CONTROL DATA CORP, 1985, 1986, 1987, 1988.

                  STATUS OF CONNECTED MDIs
                  NODE    CURRENT    SYSTEM
                 NUMBER    STATE     TITLE

                   089   AVAILABLE   MDI_8B

Only one MDI available, and is automatically selected.
```

When you are working from a NOS host, selecting the appropriate Command MDI can be important. Two different MDIs connected to a NOS host may provide access to networks that are not connected by CDCNET network solutions. These are called disjoint networks.

If you are uncertain about whether different MDIs service disjoint networks, use SETCM for each, in turn, and compare the displays from the DISPLAY_CATENET_ TITLES (DISCT) command for each. If the system titles are different for the two displays, the networks served by the two MDIs are disjoint. Or, you can try to send any command to the system in question; if the command completes successfully, the two systems are on the same catenet.

You can monitor disjoint networks simultaneously if you use the RETAIN parameter on the SETCM command. This parameter lets you retain your session with the current Command MDI when you create a new session with another MDI. Any operator alarms sent to either the retained or newly selected Command MDI are displayed immediately to the operator. To return to the previous session, reissue the SETCM command, naming the appropriate MDI.

Some NETOU commands are executed directly by the MDI you designated as the Command MDI. Other commands (those that execute in a remote system or systems) require you to use the SEND_COMMAND (SENC) command. Commands enclosed in a SENC command are processed at the specified remote CDCNET system(s).

NOTE

For a more detailed description of NETOU, see the CDCNET Network Operations and Analysis manual.

The SEND_COMMAND Command

The SENC command is used in NETOU on NOS/VE and NOS to package a NETOU command for delivery to a remote system or systems, where it is then executed. The format is as follows:

```
SEND_COMMAND COMMAND='command_string' SYSTEM=(list 1..15 of name)
```

The command string to be executed by the remote system(s) must be enclosed in single quotes.

The list of systems must be enclosed in parentheses (a single system name need not be enclosed in parentheses, however). Under NOS/VE, the SYSTEM parameter is required.

Under NOS, if you do not specify a system name or names, NETOU sends the command to the destination system(s) of the most recent SENC command. If you have not used the SENC command in the current NETOU session, NETOU sends the command to the Command MDI for execution.

The NOS/VE version of SENC has two additional parameters: OUTPUT and STATUS. OUTPUT lets you write command output to a specified file (the NOS-only ROUTE_COMMAND_RESPONSE (ROUCR) command effectively gives you the option of an alternative output file for your NOS-based NETOU session). STATUS is the standard NOS/VE status variable.

NETOU accepts either the default system name or a valid alias when you are sending commands to CDCNET systems. For example, if MDI_8B is a registered title for $DI_08002510008B, you could send a command to this DI with either of the following commands:

```
SENC 'command_string' $DI_08002510008B
```

or

```
SENC 'command_string' MDI_8B
```

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 1-1 shows escape sequences for unsupported ASCII characters and how these characters are represented on the console screen.

Table 1-1. 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	/=	/=
'	Aposotrophe	/*	/*
/	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

Terminal Definition Procedure (TDP) Commands

TDP commands are used in Terminal Definition Procedures to define the characteristics of terminal/workstation types and/or models. TDP procedures are initially developed by Network Administrators and are a part of the DI configuration procedures that reside on CYBER hosts.

TDPs execute separately from system configuration procedures, but they are referenced in commands in the system configuration procedures. Typically, TDPs are referenced by the DEFINE_LINE commands defining the communications lines connected to the DI.

If you need more information on how to use TDPs, see the CDCNET Configuration Guide.

Terminal User Procedure (TUP) Commands

TUP commands are software procedures that end users write to modify the conventions specified in the TDP. With TUPs, you can save operating conventions on an ongoing basis. Alternatively, you can issue interactive commands to tailor your terminal conventions session-by-session.

If you need more information on how to use TUPs, see the CDCNET Configuration Guide.

Load Procedure Commands

Load procedure commands define information to be downloaded to a batch device, such as a printer, or translate data to or from a device. Currently, CDCNET supports the following types of load procedures:

- Initializaton Procedures (IPs)
- File Prefix Procedures (FPs)
- Code Set Procedures
- Vertical Format Unit (VFU) Load Procedures (VLPs)

If you need more information on how to use load procedures, see the CDCNET Configuration Guide.

Configuration File Management Commands

These commands are used to access CDCNET configuration procedures such as TDP and TUP. If you need more information on how to use these procedures, see the CDCNET Configuration Guide.

Exception File Commands

Exception file commands specify the load process procedures for specific systems or the default entries, which specify the load process procedures, for systems that do not have an explicit system entry of their own. See the CDCNET Configuration Guide for more information on exception file commands.

Network Performance Analyzer (NPA) Commands

NPA is a network analysis tool made up of flexible modular software components resident in your CONTROL DATA® CYBER 170 or CYBER 180 Computer System. NPA helps you analyze the performance of your network by producing a variety of reports. These reports allow you to:

- Identify the configuration of your network
- Identify actual and potential hardware and software failures
- Identify potential congestion on communication lines
- Determine if your network is performing correctly
- Evaluate network use

You may generate these reports individually to reflect a specific aspect of the network's performance, or you may generate a group of reports that reflect an overall picture of the network. You may also create reports that are tailor-made for your specific needs. The standard set of reports is described in the CDCNET Network Operations and Analysis manual.

If you need more information on how to use NPA, see the CDCNET Network Operations and Analysis manual.

DI Dump Analyzer Commands

Under default operating conditions, a CDCNET DI dumps its memory to a host-resident file whenever it is forced to reset as the result of an error condition (refer to DI Reset Codes appendix for a complete list of reset codes). The Device Interface Dump Analyzer is a host-resident tool that reformats the information in DI dump files.

Dump Analyzer subcommands let you display information about the conditions that existed at the time of the reset, including:

- Important data structures
- Contiguous memory
- Program call chains
- Task control information

If you need more information on how to use the Dump Analyzer, see the CDCNET Network Operations and Analysis manual.

Session Control Commands 2

Session Control Commands 2

This chapter provides complete descriptions of all session control commands for both NOS/VE and NOS-based network operations environments. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operational environments.

NOTE

Session control commands are not contained within a SEND_COMMAND.

ACTIVATE_ALARMS (ACTA)
NETOU Command

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 (NOS/VE only)

Parameters

GROUPS or *GROUP (G)*

Specifies the names of the alarm groups for which alarms are to be collected. Default 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 is $OUTPUT.

STATUS (NOS/VE only)

See basic status concepts for NOS/VE System Command Language in the NOS/VE System Usage manual.

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 NETOU prolog.

Examples

```
NOU/activate_alarms

NOU/
```

CHANGE_ALARM_ENVIRONMENT (CHAAE) (NOS Only) NETOU Command

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.

If you disable receipt of alarms from a specific system, then you 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 (DC)

The community title or titles from which receipt of alarms is to be disabled. Entry of a title already disabled is permitted. Currently, the only allowed value for this parameter is CATENET, which specifies all the DIs in the catenet.

ENABLE_COMMUNITY (EC)

The community title or titles from which receipt of alarms is to be enabled. Entry of a title already enabled is permitted. Currently, the only allowed value for this parameter is CATENET, which specifies all the DIs in the catenet.

Responses Alarm environment updated.

--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)
NETOU Command

Purpose Terminates receipt of alarms from CDCNET DIs.

Format

DEACTIVATE_ALARMS
STATUS = status variable (NOS/VE only)

Parameters *STATUS* (NOS/VE only)

See basic status concepts for NOS/VE System Command Language in the NOS/VE System Usage manual.

Responses Alarms deactivated.

--ERROR-- Alarms not active.

Examples

```
deactivate_alarms

NOU/
```

DISPLAY_ALARM_ENVIRONMENT (DISAE) (NOS Only) NETOU Command

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. Currently, the only community that is displayed is CATENET, and the only domain of control supported is the catenet.

Format **DISPLAY_ALARM_ENVIRONMENT**

Parameters None.

Responses Alarm Environment.
(See example.)

Examples

```
display_alarm_environment

Alarm Environment
  Community                Alarm Status
CATENET                    Enabled

  Disabled Systems
-None-
```

DISPLAY_ALARM_HISTORY (DISAH) (NOS Only) NETOU Command

Purpose Displays alarms received at your operations station since the start of your command session, in chronological order. 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 nonblank lines of text.)

Format

DISPLAY_ALARM_HISTORY
DISPLAY_OPTION = keyword value

Parameters *DISPLAY_OPTION (DO)*

Specifies how many alarms are displayed.

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
(Followed by a 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) (NOS Only) NETOU Command

Purpose Displays the system, community, and 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 (DO)*

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. Searches the entire catenet for the titles.
COMMUNITY (C)	Displays titles of DI communities. The only supported value is CATENET.
INTERNAL_SERVICE (IS)	Displays titles of services that are known only to network services and the command MDI. Displays only the contents of the command MDI/MTI directory. 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. Displays only the contents of the command MDI/MTI directory. 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.

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) (NOS Only) NETOU Command

Purpose Displays parameters and parameter syntax information for a specified session command. The specified command must be one of the available session commands.

NOTE

On NOS/VE, this is a standard system command.

Format

```
DISPLAY_COMMAND_INFORMATION
   COMMAND = name
```

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.

Remarks The DISPLAY_COMMAND_INFORMATION session command is similar to the network command of the same name, with two important differences:

- The session command displays only the session commands available to the operator.
- The session command is not embedded within the SEND_COMMAND network command.

Examples

```
display_command_information command=include_file

file,fname = $required
username, unname = $optional
```

DISPLAY_COMMAND_LIST (DISCL)
NETOU Command

Purpose On NOS, this command displays the list of session control commands for which you are validated.

On NOS/VE, this command returns the system command list established for the operator. This list includes the NETWORK_OPERATOR_UTILITY.

Format **DISPLAY_COMMAND_LIST**

Parameters None.

Responses NOS/VE, system commands list.

NOS, alphabetical list of NETOU commands (see example).

Examples The following example shows a response generated by NOS NETOU.

```
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
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
** command to exit SENCS mode
*  escape character in SENCS mode
```

DISPLAY_COMMAND_LIST_ENTRY (DISCLE) NETOU Command

Purpose On NOS, this command displays the list of session control commands for which you are validated. This command is an alias for the DISPLAY_COMMAND_LIST command.

On NOS/VE, this command returns the list of session control commands.

Format **DISPLAY_COMMAND_LIST_ENTRY**

Parameters None.

Responses NOS/VE, list of session control commands (see example).

NOS, alphabetical list of session control commands.

Examples The following example shows a NOS/VE command list.

```
display_command_list_entry

activate_alarms                deactivate_alarms
quit                           send_command
```

DISPLAY_CONNECTED_MDI (DISCM) (NOS Only) NETOU Command

Purpose Displays the coupler numbers, system titles, and operational status of the MDIs and MTIs that are 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 MDI or MTI connected to the host.

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) (NOS Only) NETOU Command

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.

NOTE

On NOS/VE, use the standard system command INCLUDE_FILE.

Format

```
EXECUTE_COMMAND_FILE
   FILE = name
   USER_NAME = name
```

Parameters

FILE (F)

Specifies the name of the file in your catalog or in the catalog of the alternate user name, specified by the USER_NAME parameter (described next). 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 catalog other than your own, you must specify this parameter.

Remarks

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 nondisplay 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, semiprivate, 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 (NOS Only)
NETOU Command

Purpose Displays a list of CDCNET network commands for which you are validated. The commands are arranged in alphabetical order. Only the long form of the command is returned.

NOTE

On NOS/VE, HELP is a standard system command.

Format HELP

Parameters None.

Responses Displays an alphabetical list of network operator's commands (see example).

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
diplay_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
**  command to exit SENCS mode
*   escape character in SENCS mode
```

INCLUDE_FILE (INCF) (NOS Only) NETOU Command

Purpose

Directs NETOU to read the named file in your catalog. This command is an alias for the EXECUTE_COMMAND_FILE 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.

NOTE

On NOS/VE, INCF is a standard system command.

Format

INCLUDE_FILE
FILE = name
USER_NAME = name

Parameters

FILE (F)

Specifies the name of the file in your catalog or in the catalog of the alternate user name, specified by the USER_NAME parameter (described next). 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

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 nondisplay 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, the file must be public, semiprivate, 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 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 (QUI)
NETOU Command

Purpose Terminates the Network Operator Utility (NETOU) session. After the QUIT command executes, NETOU commands are not valid during a command entry session.

Format **QUIT**

Parameters None.

Examples

```
quit
```

Remarks The following NOS-only 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) (NOS Only) NETOU Command

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

Parameters None.

Responses Alarm environment restored.

Remarks Use this command after your alarm environment has been changed by the CHANGE_ALARM_ENVIRONMENT or DEACTIVATE_ALARMS command to return to the original set of DIs that report alarms to you.

Examples

```
restore_alarm_environment

Alarm environment restored.
```

ROUTE_ALARM (ROUA) (NOS Only) NETOU Command

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 are 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 fails. Both command responses and alarms may be routed to the same file.

Format

ROUTE_ALARM
FILE = list 1..2 of name and/or *DISPLAY*

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 defines the file. If the file already exists, NETOU appends subsequent alarms to the end of the existing file.

The keyword 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 are 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) (NOS Only) NETOU Command

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 are 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_COMMAND_RESPONSE
FILE = list 1..2 of name or *DISPLAY*

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 defines 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_command_response file=(todlog;display)

Command responses routed to DISPLAY and TODLOG.
```

SEND_COMMAND (SENC)
NETOU Command

Purpose Sends a CDCNET command to a DI or list of DIs.

Format **NOS Only**

```
SEND_COMMAND
   COMMAND = string
   SYSTEM = list 1..15 of name
```

NOS/VE Only

```
SEND_COMMAND
   COMMAND = string
   SYSTEM = list of 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) (NOS/VE only)

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. If no response is received in 2 minutes, a timer terminates the command.

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.

OUTPUT (O) (NOS/VE only)

The file to which a normal command response is written. Default value is $OUTPUT.

STATUS (NOS/VE only)

See basic status concepts for NOS/VE System Command Language in the NOS/VE System Usage manual.

Examples

```
senc c='dishs',s=mdi83

send_command c='write_terminal_message,..
m=''Engineering''''s network will be down until 10:00''',..
s=tdil
```

SEND_COMMAND_SEQUENCE (SENCS) (NOS Only) NETOU Command

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 special type of command entry mode (SENCS mode). You receive a prompt after entering the SENCS command: /SENCS. All commands you enter following the prompt are 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.

NOTE

When you are within SENCS mode, READY.. is the prompt at a K-display system console. SENCS/ is the prompt at an operator console.

The SENCS command may be included in a prolog or command file. If so, all subsequent commands are 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 want to send a network command to other systems while in the SENCS mode, use an escape character (a single asterisk *). To use the escape character, type the escape character * followed by the network command on the same line. This one command is then 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 is treated as part of the command text.

Format

```
SEND_COMMAND_SEQUENCE
   SYSTEM = list 1..15 of name
```

Parameters

SYSTEM (S)

Specifies 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.

--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) (NOS Only) NETOU Command

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 or not the operations session should be terminated with the old MDI (using the RETAIN parameter).

You receive responses only from the currently selected MDI. However, you 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, you 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 selects 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. 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 displays 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. Default is NO.

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 (NOS Only)
NETOU Command

Purpose Terminates the SEND_COMMAND_SEQUENCE execution mode.

Format **

Parameters None.

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/**
```

ADD Commands 3

ADD Commands 3

This chapter provides complete descriptions of all CDCNET Network Operations ADD commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

ADD_ARP_REPORTED_NETS (ADDARN)
DI Configuration Procedure/NETOU Command

Purpose

Informs Address Resolution Protocol (ARP) when the DI configured with this command responds to ARP requests for IP networks accessible by this DI. ADDARN applies only in a multi-LAN IP network environment. ADDARN switches ARP from a mode in which a DI responds to ARP requests from all IP networks/subnets to a mode in which the DI responds to ARP requests from IP networks/subnets defined by ADDARN. ADDARN limits the functionality of ARP proxy to prevent inefficient routing and broadcast loops.

ADDARN is not required to configure ARP. Omitting it informs ARP to answer ARP requests received on all Ethernets to which it connects.

Format

ADD_ARP_REPORTED_NETS
SOURCE_IP_NETWORK = list 1..4 of 0..255
DESTINATION_IP_NETWORK = list 1..32 of list 1..4 of 0..255 or *keyword value*

Parameters

SOURCE_IP_NETWORK (SIN)

ARP requests received from this specified IP network/subnet are eligible for a reply if the target IP address is on one of the specified destination IP networks. (See DESTINATION_IP_NETWORK parameter.) The format is similar to the decimal octet convention used by TCP/IP, except 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).

DESTINATION_IP_NETWORK (DIN)

List of IP networks/subnets reportable via an ARP reply. Specify this parameter as a list, or use keyword values NONE or ALL.

Keyword Value	Description
NONE	ARP proxy disabled. Only ARP requests with a target IP address on the SOURCE_IP_NETWORK are considered for a reply.
ALL	ARP proxy in effect for all IP networks/subnets.

Default is ALL.

Responses

Reportable IP network/subnet is added for source IP network/subnet <source_ip_network>.

--ERROR-- Reportable IP network/subnet is already defined for source IP network <source_ip_network>.

--ERROR-- IP network <source_ip_network> is not defined.

--ERROR-- IP subnet <source_ip_network> is not defined.

--ERROR-- IP network/subnet <source_ip_network> is not directly connected.

--FATAL-- Not enough memory is currently available for required table space.

Remarks This command is useful to prevent ARP looping and to reduce Ethernet traffic.

Examples The following example shows how to use this command. Given the configuration in figure 3-1, if the source host sends IP datagrams to a destination host, a potential ARP broadcast loop can occur. ADD_ARP_REPORTED_NETS can prevent the ARP broadcast loop.

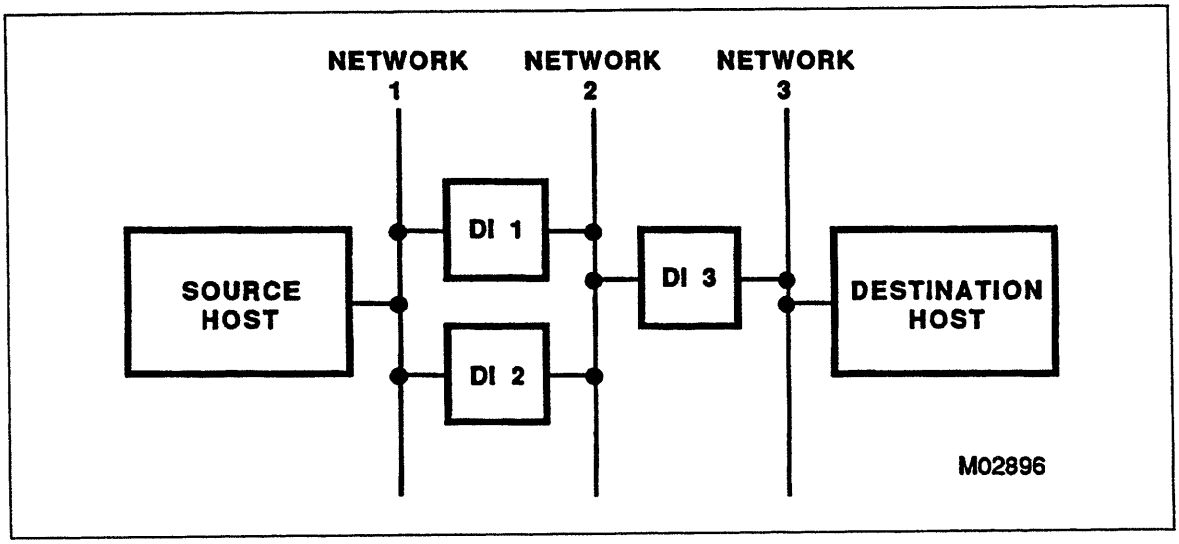


Figure 3-1. ADD_ARP_REPORTED_NETS Example

The source host sends an ARP request on network 1 to determine the destination host's Ethernet address. DI 1 and DI 2 propagate the ARP request on network 2 to search for network 3 (the network where the destination host resides). DI 3 sees both ARP requests and replies to both DI 1 and DI 2, since it directly connects to network 3. However, DI 1 and DI 2 see each others' ARP request and propagate it on network 1 searching for network 3. This cycle continues until the instigating ARP request is answered.

If one of the DIs connecting networks 1 and 2 is disabled to respond to ARP requests for network 3, the broadcast loop is avoided. This is done by entering the following commands in DI 2's configuration file:

```
add_arp_reported_nets ..
  source_ip_network=1 ..
  destination_ip_network=((1),(2))

add_arp_reported_nets ..
  source_ip_network=2 ..
  destination_ip_network=((1),(2))
```

These commands inform DI 2 to process ARP requests received on either of its connected Ethernet networks only for target networks 1 and 2. This enables DI 1 to propagate only the ARP request on network 2. The source host can use either DI 1 or DI 2 to reach the destination host.

ADD_CLNS_FILTER (ADDCF)
NETOU Command

Purpose

Creates a CLNS filter entry for the specified network. As CLNS datagrams are received on the specified network, the source and destination NSAP addresses are compared to CLNS filter entries. For the addresses to match, only the address prefix portion must match. For the entry to be used, both the source and destination addresses must match. The CLNS filter entry specifies whether the matching CLNS datagram is to be routed or discarded.

During filter processing, CLNS filter entries with longer source address prefixes are tried first. For entries with equal source address lengths, the longer destination address prefixes are used first. In this way, CLNS filter entries with more specific (that is, longer) NSAP address prefixes will be used sooner. The length of address prefixes specified as ANY are considered to be zero.

Format

ADD_CLNS_FILTER
NETWORK_NAME = name
SOURCE_ADDRESS_PREFIX = string 2..40 or ANY
DESTINATION_ADDRESS_PREFIX = string 2..40 or ANY
ACCESS_ALLOWED = boolean

Parameters

NETWORK_NAME (NN)

Identifies the name of a network defined in this system. This is the network to which the CLNS filter entry is added.

SOURCE_ADDRESS_PREFIX (SAP)

Specifies the NSAP address prefix used to match source addresses of datagrams routed through the network. A source address match occurs if any leading portion (including all) of the source NSAP address from the datagram matches this value. ANY specifies that a match occurs for any CLNS datagram whose destination address matches this CLNS filter entry. The elements of the string defining an address prefix must be hexadecimal digits. An address prefix must include an even number of hexadecimal digits.

DESTINATION_ADDRESS_PREFIX (DAP)

Specifies the NSAP address prefix used to match destination addresses of datagrams routed through the network. A destination address match occurs if any leading portion (including all) of the destination NSAP address from the datagram matches this value. ANY specifies that a match occurs for any CLNS datagram whose source address matches this CLNS filter entry. The elements of the string defining an address prefix must be hexadecimal digits. An address prefix must include an even number of hexadecimal digits.

ACCESS_ALLOWED (AA)

Specifies whether a CLNS datagram matching this entry is routed. If TRUE, the datagram is forwarded. If FALSE, the datagram is discarded. The default is TRUE.

Responses

The CLNS filter entry is added to network <network_name> as follows:

Source NSAP address prefix	<source_NSAP_address_prefix>
Destination NSAP address prefix	<destination_NSAP_address_prefix>
Access allowed	<access_allowed>

--ERROR-- Network <network_name> is not defined.

--ERROR-- This CLNS filter definition is already defined.

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

--ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_clns_filter network_name=NETWORK1 source_address_prefix='00470023' ..
  destination_address_prefix='0032004556' access_allowed=TRUE

 The CLNS filter entry is added to network NETWORK1 as follows

 Source NSAP address prefix          00470023
 Destination NSAP address prefix     0032004556
 Access allowed                      TRUE


add_clns_filter network_name=NETWORK2 source_address_prefix='00470023' ..
  destination_address_prefix='0032004556' access_allowed=TRUE

 --ERROR-- Network NETWORK2 is not defined.


add_clns_filter network_name=NETWORK1 source_address_prefix='00470023' ..
  destination_address_prefix='0032004556' access_allowed=TRUE

 --ERROR-- This CLNS filter definition is already defined.


add_clns_filter network_name=NETWORK2 source_address_prefix='004700FG' ..
  destination_address_prefix='0032004556' access_allowed=TRUE

 --ERROR-- The NSAP address prefix 004700FG contains non-hexadecimal
 characters.
```

ADD_CONS_INCALL (ADDCI)
DI Configuration Procedure/NETOU Command

Purpose Defines a CONS incall definition. An incall definition is from the perspective of the CDCNET network. The incall information controls X.25 calls coming into the CDCNET network destined for CONS.

The source NSAP address associates the incoming call with the incall definition. The source NSAP address is either specified in an X.25 facility (the Address Extension Facility) for a 1984 PDN or is obtained from the static mapping table defined by ADD_CONS_1980_X25_MAPPING for a 1980 PDN.

Format

ADD_CONS_INCALL
ADDRESS_PREFIX = string 2..40
ACCEPT_PDN_CHARGES = boolean

Parameters

ADDRESS_PREFIX (AP)

Specifies the address prefix of the peer's NSAP address (the source NSAP address) for this incall definition. The address prefix can be any leading portion of the NSAP address. The address prefix can associate the incall definition with multiple NSAP addresses. If multiple ADD_CONS_INCALL definitions exist that match a given NSAP address, the call definition associated with the longest string is used. The elements of the string you enter must contain an even number of hexadecimal digits.

ACCEPT_PDN_CHARGES (APC)

Specifies how calls initiated from the peer CONS user should be handled. If TRUE, a call is accepted with either normal or reverse charging. If FALSE, a call is accepted with only normal charging. Default is FALSE.

Responses The information for CONS incall definition is added as follows:

NSAP address prefix = <address_prefix>
Accept PDN charges = <accept_pdn_charges>

--ERROR-- The CONS incall definition is already defined as follows:

NSAP address prefix = <address_prefix>
Accept PDN charges = <accept_pdn_charges>

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

--ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_cons_incall address_prefix='00470023' accept_pdn_charges=TRUE

   The CONS incall definition is added as follows:

   NSAP address prefix     00470023
   Accept PDN charges      TRUE

add_cons_incall address_prefix='00470023' accept_pdn_charges=FALSE

 --ERROR-- The CONS incall definition is already defined as

NSAP address prefix     00470023
Accept PDN charges      TRUE

add_cons_incall address_prefix='004700FG' accept_pdn_charges=TRUE

 --ERROR-- The NSAP address prefix 004700FG contains non-hexadecimal characters.
```

ADD_CONS_OUTCALL (ADDCO)
DI Configuration Procedure/NETOU Command

Purpose

Defines a CONS outcall definition. An outcall definition is from the perspective of the CDCNET network. CDCNET uses the outcall information to generate the proper CONS/X.25 call requests to the public data network.

The destination NSAP address associates the outgoing call with the outcall definition. A CONS user internally specifies the destination NSAP address. Indirectly, the destination NSAP address may have been configured by the NAM/VE command ADD_OSI_ADDRESS or obtained by directory services.

There are multiple facility parameters. In addition to the facilities that can be added by these parameters, there are other facilities that can be added by the CONS entity. For 1980 and 1984 PDNs, the packet size facility can be added, based on the setting of the SUBSCRIBED_FACILITIES parameter of DEFINE_X25_INTERFACE. For 1984 PDNs, the expedited data negotiation facility (with value FALSE) and the called and calling address extension facilities (with values based on the corresponding NSAP addresses) are added.

When you specify two or more categories, facility markers (two octets in length) are automatically inserted between each of the categories of facilities (X25_FACILITIES, LOCAL_NON_X25_FACILITIES, REMOTE_NON_X25_FACILITIES, and CCITT_DTE_FACILITIES).

Format

ADD_CONS_OUTCALL
ADDRESS_PREFIX = string 2..40
REMOTE_DTE_ADDRESS = string 1..15
LOCAL_DTE_ADDRESS = string 1..15
ADD_REVERSE_CHARGES = boolean
ADD_FAST_SELECT = boolean
INTERFACE_STANDARD = key
X25_FACILITIES = string 2..218
LOCAL_NON_X25_FACILITIES = string 2..218
REMOTE_NON_X25_FACILITIES = string 2..218
CCITT_DTE_FACILITIES = string 2..218
USER_DATA = string 2..256

Parameters

ADDRESS_PREFIX (AP)

Specifies the address prefix of the peer's NSAP address (the destination NSAP address) for this incall definition. The address prefix can be any leading portion of the NSAP address. The address prefix can associate the outcall definition with multiple NSAP addresses. If multiple ADD_CONS_INCALL definitions exist that match a given NSAP address, the call definition associated with the longest string is used. The elements of the string you enter must contain an even number of hexadecimal digits.

REMOTE_DTE_ADDRESS (RDA)

Specifies the X.25-called DTE address used when a virtual circuit is created in order to transmit data to the associated NSAP address. Specify this parameter as a string of digits.

For NSAP addresses where the next hop trunk is an X.25 trunk, REMOTE_DTE_ADDRESS must be specified on the ADD_NON_CDC_ROUTING_ENTRY or ADD_CONS_OUTCALL commands. If specified by both, the ADD_CONS_OUTCALL command takes precedence. If specified by neither, any CONS connect request to the NSAP is rejected. If the next hop trunk is an X.25 trunk identified internally by dynamic routing by an X.25 network solution, the other remote DTE address is the one defined by DEFINE_X25_NET, not ADD_NON_CDC_ROUTING_ENTRY. The default for Ether (LLC_2) and HDLC trunks is that the called DTE address is zero length.

LOCAL_DTE_ADDRESS (LDA)

Specifies the X.25 calling DTE address used when creating a virtual circuit to transmit data to the associated NSAP address. Specify this parameter as a string of digits 0 through 9.

For NSAP addresses where the next hop trunk is an X.25 trunk, LOCAL_DTE_ADDRESS can be specified on either the DEFINE_X25_INTERFACE or ADD_CONS_OUTCALL commands. If specified by both, the ADD_CONS_OUTCALL command takes precedence. If specified by neither, any CONS connect request to the NSAP address has a zero length calling DTE. The default for Ether (LLC_2) and HDLC trunks is that the calling DTE address is zero length.

ADD_REVERSE_CHARGES (ARC)

Specifies whether to use the reverse charges facility for outgoing calls. If TRUE, the reverse charges facility is included in outgoing calls. If FALSE, reverse charging will not be requested. The default is FALSE.

ADD_FAST_SELECT (AFS)

Specifies whether to use the fast select facility for outgoing calls. If TRUE, the fast select facility is included in outgoing calls. If user data is greater than 16 octets, this parameter is automatically set to TRUE. Otherwise, the default is FALSE.

INTERFACE_STANDARD (IS)

Specifies which X.25 standard to use when formatting the call for a given trunk.

SAME_AS_INTERFACE

Specifies that CONS use the value defined for the actual trunk by the DEFINE_X25_INTERFACE command (could be either CCITT_1980 or CCITT_1984).

CCITT_1980

Specifies that CONS formats the call using the 1980 CCITT standard no matter what value the DEFINE_X25_INTERFACE command specifies. A CCITT_1980 call does not include certain facilities (expedited data negotiation and address extension) that are present in a CCITT_1984 call.

The default is SAME_AS_INTERFACE.

X25 _FACILITIES (XF)

Specifies X.25 facilities options as defined by the X.25 CCITT protocol. This parameter is specified as an even-numbered string of hexadecimal digits. There is no additional validation performed on the string other than to check that the length of all the facilities plus implied markers specified by this command does not exceed 109 octets. The default is that no additional facilities are generated for this category.

LOCAL _NON _X25 _FACILITIES (LNXF)

Specifies non-X.25 facilities options provided by the local network as allowed by the X.25 CCITT protocol. This parameter is specified as an even-numbered string of hexadecimal digits. There is no additional validation performed on the string other than to check that the length of all the facilities plus implied markers specified by this command does not exceed 109 octets. The default is that no facilities are generated for this category.

REMOTE _NON _X25 _FACILITIES (RNXF)

Specifies non-X.25 facilities options provided by the remote network as allowed by the X.25 CCITT protocol. This parameter is specified as an even-numbered string of hexadecimal digits. There is no additional validation performed on the string other than to check that the length of all the facilities plus implied markers specified by this command does not exceed 109 octets. The default is that no facilities are generated for this category.

CCITT _DTE _FACILITIES (CDF)

Specifies CCITT DTE facilities options as defined by the X.25 CCITT protocol. This parameter is specified as an even-numbered string of hexadecimal digits. There is no additional validation performed on the string other than to check that the length of all the facilities plus implied markers specified by this command does not exceed 109 octets. The default is that no additional facilities are generated for this category.

USER _DATA (UD)

Specifies an even-numbered string of hexadecimal digits. This parameter value overrides any user data specified by the CONS user (such as the user data propagated from TP0 as specified on the CHANGE_OSI_ TRANSPORT command) and places the string into the USER_DATA field of the X.25 call. The default is to use the user data specified by the CONS user.

Responses The CONS outcall definition is added as follows:

NSAP address prefix = <address_prefix>
Remote DTE address = <remote_dte_address>
Local DTE address = <local_dte_address>
Add reverse charging = <add_reverse_charges>
Add fast select = <add_fast_select>
Interface standard = <interface_standard>
X.25 facilities = <x25_facilities>
Local non-X.25 facilities = <local_non_x25_facilities>
Remote non-X.25 facilities = <remote_non_x25_facilities>
CCITT DTE facilities = <ccitt_dte_facilities>
User Data = <user_data>

--ERROR-- The CONS outcall definition is already defined as follows:

NSAP address prefix = <address_prefix>
Remote DTE address = <remote_dte_address>
Local DTE address = <local_dte_address>
Add reverse charging = <add_reverse_charges>
Add fast select = <add_fast_select>
Interface standard = <interface_standard>
X.25 facilities = <x25_facilities>
Local non-X.25 facilities = <local_non_x25_facilities>
Remote non-X.25 facilities = <remote_non_x25_facilities>
CCITT DTE facilities = <ccitt_dte_facilities>
User Data = <user_data>

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters. --ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

--ERROR-- DTE address <string> contains non-decimal characters.

--ERROR-- Facilities <string> contains non-hexadecimal characters.

--ERROR-- Facilities <string> contains an odd number of digits.

--ERROR-- Specified facilities total length of <number> octets exceeds the maximum of 109 octets.

--ERROR-- User_data <string> contains non-hexadecimal characters.

--ERROR-- User_data <string> contains an odd number of digits.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_cons_outcall address_prefix='00470023' remote_dte_address='123456789' ..
  add_reverse_charges=TRUE remote_non_x25_facilities='112233'

    The CONS outcall definition is added as follows:

    NSAP address prefix          00470023
    Remote DTE address           123456789
    Local DTE address            None specified
    Add reverse charging         TRUE
    Add fast select              FALSE
    Interface standard           SAME_AS_INTERFACE
    X.25 facilities              None specified
    Local non-X.25 facilities    None specified
    Remote non-X.25 facilities   112233
    CCITT DTE facilities         None specified
    User Data                    CONS user specified

add_cons_outcall address_prefix='00470023' add_fast_select=TRUE ..
  remote_non_x25_facilities='998877'

 --ERROR-- The CONS outcall definition is already defined as follows:

    NSAP address prefix          00470023
    Remote DTE address           123456789
    Local DTE address            None specified
    Add reverse charging         TRUE
    Add fast select              FALSE
    Interface standard           SAME_AS_INTERFACE
    X.25 facilities              None specified
    Local non-X.25 facilities    None specified
    Remote non-X.25 facilities   112233
    CCITT DTE facilities         None specified
    User Data                    CONS user specified

add_cons_outcall address_prefix='004700FG' user_data='03010100'

 --ERROR-- The NSAP address prefix 004700FG contains non-hexadecimal characters.
```

ADD_CONS_1980_X25_MAPPING (ADDC1XM)
DI Configuration Procedure/NETOU Command

Purpose

Configures the information incoming from an X.25 PDN needed to route CONS/TP0 traffic to the proper destination.

ADD_CONS_1980_X25_MAPPING defines the following information:

- The destination NSAP address
- The destination (called) DTE address
- The source NSAP address
- The source (calling) DTE address

The ADDC1XM command can be entered repeatedly. This may be necessary if more than one destination DTE/NSAP address and source DTE/NSAP address association exists.

Format

ADD_CONS_1980_X25_MAPPING
DESTINATION_NSAP_ADDRESS = string 2..40
DESTINATION_DTE_ADDRESS = string 1..15
SOURCE_NSAP_ADDRESS = string 2..40
SOURCE_DTE_ADDRESS = string 1..15

Parameters

DESTINATION_NSAP_ADDRESS (DNA)

Specifies the destination NSAP address used for all X.25 Incoming Call packets containing called and calling DTE addresses matching the values established by the DESTINATION_DTE_ADDRESS and SOURCE_DTE_ADDRESS parameters of this command. These Call packets cause a corresponding CONS connection to be made to the destination NSAP address specified by this parameter.

The elements of the string you enter must contain an even number of hexadecimal digits.

DESTINATION_DTE_ADDRESS (DDA)

Specifies the destination DTE address used for mapping the called DTE address of X.25 Incoming Call packets to the destination NSAP address specified by the DESTINATION_NSAP_ADDRESS parameter of this command. Default is no called DTE address (Incoming Call packets that do not contain a called DTE address are considered matching).

The called and calling DTE addresses of Incoming Call packets must match the values established by the SDA parameter and this parameter before the SDAP addresses specified by this command occur.

The elements of the string you enter must contain decimal digits.

SOURCE_NSAP_ADDRESS (SNA)

Specifies the source NSAP address used for all X.25 Incoming Call packets containing called and calling DTE addresses matching the values established by the DESTINATION_DTE_ADDRESS and SOURCE_DTE_ADDRESS parameters. These Call packets cause a corresponding CONS connection to be made to the NSAP address specified by the DESTINATION_NSAP_ADDRESS parameter. The source NSAP address used for the CONS connection is the value specified by this parameter. The default for this parameter is the NSAP address of this system.

The elements of the string you enter must contain an even number of hexadecimal digits.

If an entry is given in which no source DTE address is supplied but a source NSAP is supplied, and an incoming call contains no source DTE address but a destination DTE address that matches this entry, then the entry is considered found and the supplied NSAP address is used instead of the default.

If the call source may be from two different systems, the correct NSAP address should be entered because the Transport protocol, which states the source NSAP address along with the source Transport reference number, must be unique. If they are not unique, unexpected results such as disconnects or hung connections may occur.

Also, the destination end system may require the correct source NSAP address for validation purposes. For example, with the RTS (Reliable Transfer Service) application on at the NOS/VE host, the incoming call is validated by the source NSAP address. The source NSAP address given to the RTS application must match the source NSAP address given on this command.

For the RTS application on the NOS/VE host, the command, ADD_OSI_ADDRESS (ADDOSIA), must be executed for the NSAP address of the incoming call destination. For RTS to pass the incoming validation check, the address given to ADDOSIA must match the source NSAP address. No such requirement exists for FTAM.

SOURCE_DTE_ADDRESS (SDA)

Specifies the source DTE address used for mapping the calling DTE address of X.25 Incoming Call packets to the source NSAP address specified by the SNA parameter of this command. Default is no calling DTE address (Incoming Call packets that do not contain a calling DTE address are considered matching).

NOTE

Both the called and calling DTE addresses of Incoming Call packets must match the values established by the DDA parameter and this parameter before the mapping to the NSAP addresses specified by this command occurs.

The elements of the string entered must contain decimal digits.

It is not required that an incoming X.25 Connect Request from an X.25 PDN contain a calling (source) DTE address. This information is needed if the destined system needs to know the origination of the call and the destination DTE address is not enough to determine this information. For example, if this NDI expects to receive only incoming CONS/TP0 traffic over an X.25 PDN from one foreign system, then the source NSAP address can be determined from the incoming called (destination) DTE address.

If this NDI expects to receive incoming CONS/TP0 traffic over an X.25 PDN from more than one foreign system, but each foreign system is sending data to a different destination system, the source NSAP address can be determined from the destination DTE address.

If the incoming call contains a source DTE address, then it must be supplied in order for this entry to match.

Responses The information for routing incoming CONS/TP0 traffic is added as follows:

Destination NSAP address = <destination_nsap_address>
Destination DTE address = <destination_dte_address>
Source NSAP address = <source_nsap_address>
Source DTE address = <source_dte_address>

--ERROR-- The NSAP address <nsap address> contains non-hexadecimal characters.

--ERROR-- The NSAP address <nsap address> contains an odd number of digits.

--ERROR-- The DTE address <dte address> contains non-decimal characters.

--ERROR-- The information for routing incoming CONS/TP0 traffic is already defined as follows:

Destination NSAP address = <destination_nsap_address>
Destination DTE address = <destination_dte_address>
Source NSAP address = <source_nsap_address>
Source DTE address = <source_dte_address>

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
add_cons_1980_x25_mapping dda='1234567' ..
 dna='39840f01bb7b000000000000010800251000840l'..
 sda='789012'  sna='47000400030001202301100700001'

   The information for routing incoming CONS/TP0 traffic is
   added as follows:

   Destination NSAP address = 39840f01bb7b000000000000010800251000840l
   Destination DTE address = 1234567
   Source NSAP address = 47000400030001202301100700001
   Source DTE address = 789012


add_cons_1980_x25_mapping ..
 dna='39840f01bb7b000000000000010800251000840l'

   The information for routing incoming CONS/TP0 traffic is
   added as follows:

   Destination NSAP address = 39840f01bb7b000000000000010800251000840l
   Destination DTE address = Not Specified
   Source NSAP address = 39840f01bb7b000000000000010800251000720l
   Source DTE address = Not Specified


addc1xm  dda='1234567'  dna='000000000800251000 8G'

 --ERROR-- The NSAP address 000000000800251000 8G
contains non-hexadecimal characters.


addc1xm dda='12345FF'   dna='000000000800251000 8G'

 --ERROR-- The DTE address 12345FF contains non-decimal characters.


addc1xm dda='1234567'  ..
 dna='39840f01bb7b000000000000010800251000840l'

 --ERROR-- The information for routing incoming
CONS/TP0 traffic is already defined as follows:

Destination NSAP address = 39840f01bb7b000000000000010800251000840l
Destination DTE address = 1234567
Source NSAP address = 39840f01bb7b000000000000010800251000720l
Source DTE address = Not Specified
```

ADD_EGP_REPORTED_NETS (ADDERN)
DI Configuration Procedure/NETOU Command

Purpose Adds the specified IP network numbers to the list of local networks reported by Exterior Gateway Protocol (EGP) to its peers for this DI. If the list of local networks was previously empty, EGP would have reported all local networks to its peers. ADDERN switches EGP from reporting all IP networks to reporting only IP networks defined by this command.

Format

ADD_EGP_REPORTED_NETS
IP_NETWORK = list 1..32 of list 1..3 of 0..255

Parameters

IP_NETWORK (IN)

List of IP network numbers reported to EGP peers. The format is similar to the decimal octet convention used by TCP/IP, except periods are replaced with commas, and the list is enclosed in parentheses. In addition, the address is a list of three numbers instead of four. Since the address is a *network* address, the host portion is omitted. For example, the IP address 128.2.53.0 is represented as (128,2,53).

Responses IP networks are added to the reported EGP list.

--FATAL-- Not enough memory is currently available for required table space.

Remarks This command is useful for security. For example, you might want one gateway not to know about another gateway.

Examples The following example shows how to use this command. Given the configuration in figure 3-2, the best path for a DDN host to connect to a host on 192.12.251 is through DI 1. Likewise, DI 4 is the best path for a DDN host to connect to a host on 192.5.209. Since DI 1 can reach network 192.5.209, it could report to its EGP peer that 192.5.209 is reachable through it. This would cause an inefficient route to 192.5.209.

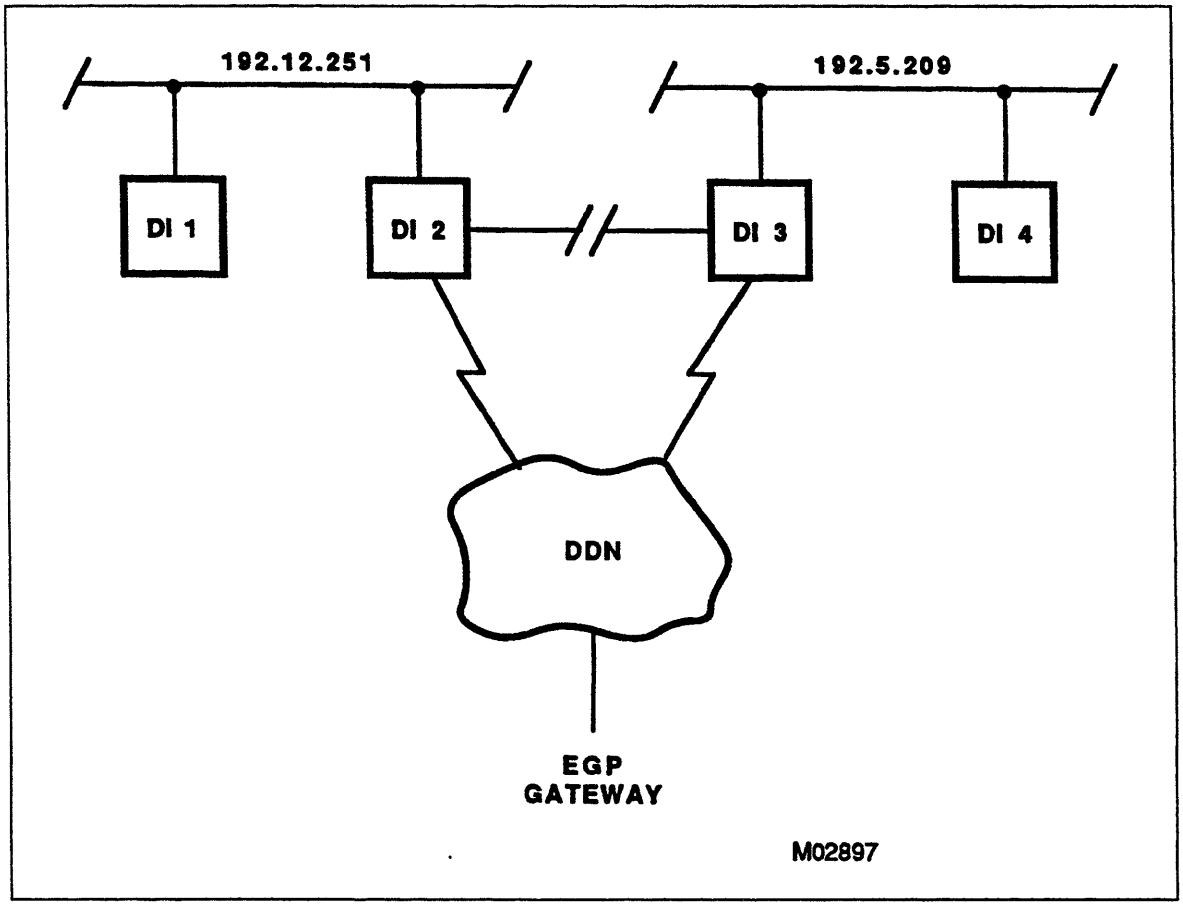


Figure 3-2. ADD_EGP_REPORTED_NETS Example

To cause each gateway to report only its directly connected Ethernet, enter the following commands in the DI 1 and DI 4 configuration files.

For DI 1:

```
add_egp_reported_nets ip_network=((192,12,251))
```

For DI 4:

```
add_egp_reported_nets ip_network=((192,5,209))
```

ADD_INCOMING_SERVICE_ACCESS (ADDISA)
DI Configuration Procedure/NETOU Command

Purpose Adds the titles of servers that may be accessed coming in via the network solution.

Format

ADD_INCOMING_SERVICE_ACCESS
TITLE = list 1..15 of name or string 1..255
NETWORK_NAME = name

Parameters

TITLE (T)

Title of a server that may be accessed coming in via the network solution. The title may contain wildcard characters to specify a set of allowed titles (see the CDCNET Network Operations and Analysis manual for information on wildcard characters).

Title is case-sensitive when entered as a string. For example, if the command defining a service specifies the service title in uppercase, it must be specified in uppercase for the ADDISA command or it is seen as a different title.

For more information on the use of service titles, see the CDCNET Configuration manual.

NETWORK_NAME (NN)

Identifies the name of a currently defined network within this system. This is the network solution upon which the accessible services are being defined. Defaults to the previously defined network when on the configuration file. It is required when the command is entered by the operator.

Responses

Incoming service access added to network <network_name>.

--WARNING-- Incoming service access added to network <network_name>.
Title <title> already added to network.
{Line repeats for each title already added to network}

--ERROR-- A network is not defined.

--ERROR-- Network <network_name> is not defined.

--ERROR-- Title <title> may not appear more than once in the list of specified titles.

--ERROR-- Expecting NAME or STRING, found <xxx>.

--FATAL-- Not enough memory is currently available for required table space.

Examples

In this example, an HDLC trunk links two separate sites (figure 3-3). Assume that service access between the two sites must be limited to the following:

- QTF/PTF file transfers in either direction.
- Site 1 has unrestricted interactive service access to site 2.
- Site 2 interactive service access to site 1 is limited to Host_1.

Add the following to the configuration file for NDI_1:

```
define_hdlc_net..
   trunk_name=hdlc_trunk_1,network_name=hdlc_net_1,..
   network_id=3..
   incoming_service_access=(file_transfer)..
   outgoing_service_access=(file_transfer,interactive)
add_incoming_service_access..
   title=host_1
```

The ADD_INCOMING_SERVICE_ACCESS command adds Host_1 as the only site 1 host to which site 2 users can make an interactive connection.

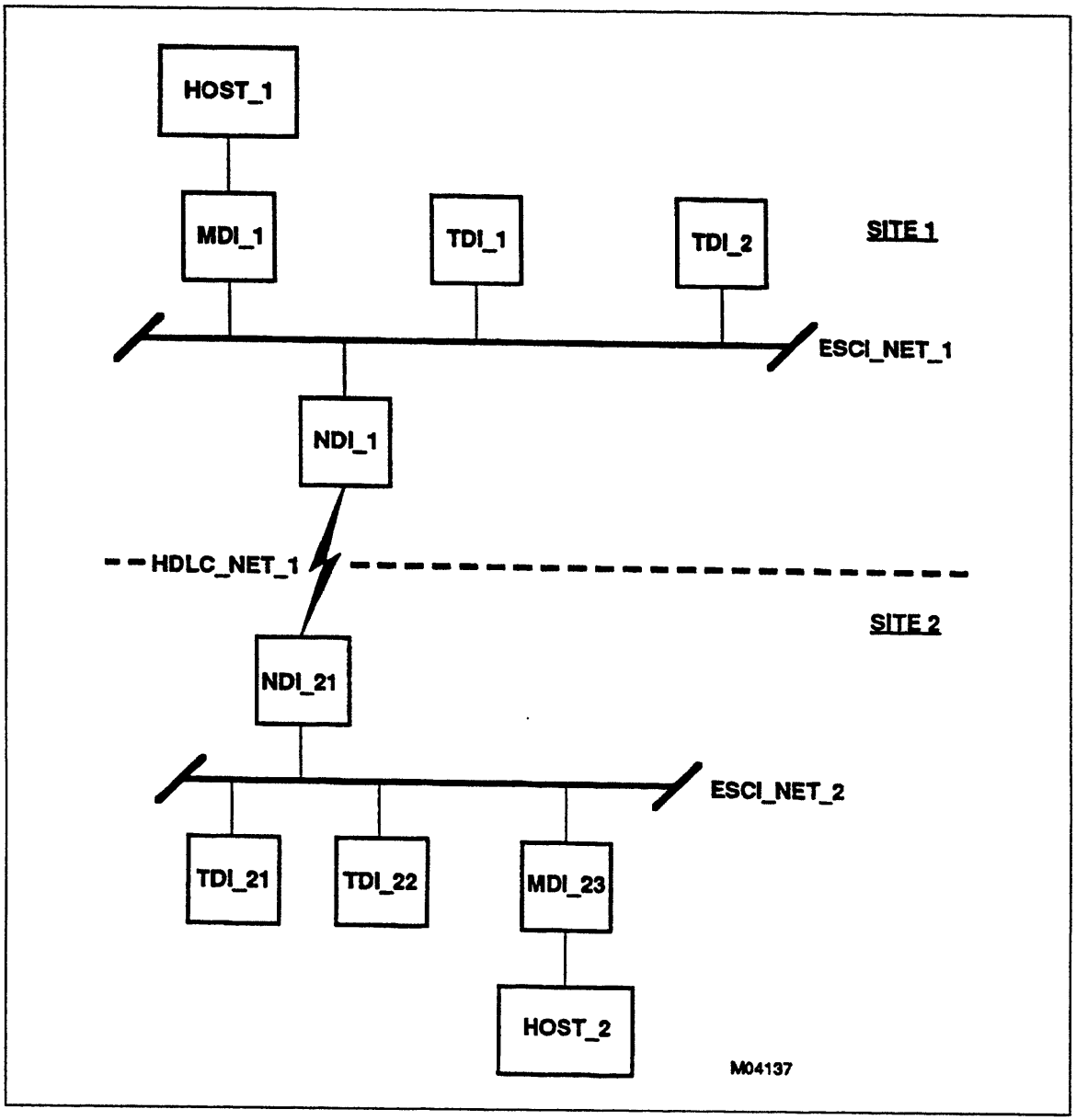


Figure 3-3. ADDISA Example

ADD_IP_FILTER (ADDIF)
NETOU Command

Purpose

Creates an IP filter entry for the specified IP network. As IP datagrams are received on the specified network, the system compares the protocol, IP addresses, and port numbers to IP filter entries. The portions of the IP addresses used are determined by the masks. When an IP datagram is processed, a particular filter entry is selected if all its parameters match the IP datagram. A value of ANY for a parameter specifies that any value for that item from the datagram is a match. The ACCESS_ALLOWED parameter from the IP filter entry specifies whether the matching IP datagram is to be routed or discarded. When a datagram is discarded, a statistics field and a MIB variable are incremented. The order of IP filter entries used is the same as the order in which they are created. The system administrator can use masks and change the order created to cause different effects with similar addresses.

Format

ADD_IP_FILTER
IP_NETWORK = list 1..4 of integer 0..255
SOURCE_IP_ADDRESS = list 1..4 of integer 0..255 or ANY
DESTINATION_IP_ADDRESS = list 1..4 of integer 0..255 or ANY
SOURCE_IP_MASK = list 1..4 of integer 0..255 or HOST
DESTINATION_IP_MASK = list 1..4 of integer 0..25 or HOST
PROTOCOL = integer 1..254 or keyword value
SOURCE_PORT_NUMBER = list 1..10 of integer 0..65535 or ANY
DESTINATION_PORT_NUMBER = list 1..10 of integer 0..65535 or ANY
ACCESS_ALLOWED = boolean
NUMBER_DATAGRAMS_LOGGED = integer 0..255

Parameters

IP_NETWORK (IN)

Identifies an IP network defined in this system. This is the network to which the IP filter entry is added.

SOURCE_IP_ADDRESS (SIA)

Specifies the IP address used to filter source IP addresses from datagrams received on this IP network. A source address match occurs if the source IP address from the datagram matches this value after applying the SOURCE_IP_MASK parameter. ANY specifies that any source IP address in the datagram satisfies this part of the match.

DESTINATION_IP_ADDRESS (DIA)

Specifies the IP address used to filter destination IP addresses from datagrams received on this IP network. A destination address match occurs if the destination IP address from the datagram matches this value after applying the DESTINATION_IP_MASK parameter. ANY specifies that any destination IP address in the datagram satisfies this part of the match.

SOURCE_IP_MASK (SIM)

Specifies which parts of the SOURCE_IP_ADDRESS parameter are compared to the IP datagram's source address to determine a source address match. List items not included default to zeros. The keyword HOST uses (255,255,255,255) to specify that the entire source IP address is compared. This parameter is not used if the SOURCE_IP_ADDRESS parameter is ANY. The default is HOST.

DESTINATION_IP_MASK (DIM)

Specifies which parts of the DESTINATION_IP_ADDRESS parameter are compared to the IP datagram's destination address to determine a destination address match. List items not included default to zeros. The keyword HOST uses (255,255,255,255) to specify that the entire destination IP address is compared. This parameter is not used if the DESTINATION_IP_ADDRESS parameter is ANY. The default is HOST.

PROTOCOL (P)

Specifies the datagram protocol filtering to use for this entry. A protocol match occurs if the protocol field in the IP datagram matches this value. The keywords ICMP, TCP, and UDP specify corresponding protocol values; ANY indicates that any protocol value matches. The default is ANY.

SOURCE_PORT_NUMBER (SPN)

Specifies ranges of source port number values used to filter the datagram's source port field received on the specified IP network. A match occurs if the source port number from the datagrams falls into any of the specified ranges. ANY specifies that any source port number from the datagram satisfies this part of the match. Only protocol TCP and UDP datagrams include source and destination port numbers. For a filter entry to match a datagram that does not include a source port number, the filter entry must specify ANY. The default is ANY.

DESTINATION_PORT_NUMBER (DPN)

Specifies ranges of destination port number values used to filter the datagram's destination port field received on the specified IP network. A match occurs if the destination port number from the datagrams falls into any of the specified ranges. ANY specifies that any destination port number from the datagram satisfies this part of the match. Only protocol TCP and UDP datagrams include source and destination port numbers. For a filter entry to match a datagram that does not include a destination port number, the filter entry must specify ANY. The default is ANY.

ACCESS_ALLOWED (AA)

Specifies whether an IP datagram matching this entry is routed. If TRUE, the datagram is forwarded. If FALSE, the datagram is discarded. The default is TRUE.

NUMBER_DATAGRAMS_LOGGED (NDL)

Specifies how many of the filtered IP datagrams should be logged in a given statistics interval. The logging occurs only if DOD_INTERNET is specified on the PROCESS parameter of the START_PROCESS_METRICS command. The log message recording the filtered datagrams is log ID 1712. The default is 0.

Responses

The IP filter entry is added to network <ip_network> as follows:

Source IP address	<source_ip_address>
Destination IP address	<destination_ip_address>
Source IP mask	<source_ip_mask>
Destination IP mask	<destination_ip_mask>
Protocol	<protocol>
Source port ranges	<source_port_number>
Destination port ranges	<destination_port_number>
Access allowed	<access_allowed>
Number datagrams logged	<number_datagrams_logged>

--ERROR-- IP network <ip_network> is not defined.

--ERROR-- The IP Filter entry is already defined on IP network <ip_network>.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_ip_filter ip_network=(129,179,52) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255 access_allowed=TRUE
  number_datagrams_logged=10

 The IP filter entry is added to IP network 129,179,56 as follows:

 Source IP address          129.179.000.000
 Destination IP address     129.179.630.000
 Source IP mask             255.255.000.000
 Destination IP mask        255.255.255.255
 Protocol                   any
 Source port ranges         1..126
 Destination port ranges    127..255
 Access allowed             TRUE
 Number datagrams logged    10


add_ip_filter ip_network=(129,179,54) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255 access_allowed=TRUE ..
  number_datagrams_logged=10

 --ERROR-- IP network 179,179,54 is not defined.


add_ip_filter ip_network=(129,179,52) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255 access_allowed=TRUE..
  number_datagrams_logged=10


 --ERROR-- The IP filter definition is already defined on
 IP network 129,179,52
```

ADD_NON_CDC_ADDRESS_PREFIX (ADDNCAP)
DI Configuration Procedure/NETOU Command

Purpose Defines information used to route traffic to a non-CDCNET OSI system. This information includes a list of address prefixes of NSAP addresses in one or more non-CDCNET systems which can be reached from the same CDCNET boundary system.

Format

ADD_NON_CDC_ADDRESS_PREFIX
ADDRESS_PREFIX = list 1..10 of string 2..40
NEXT_HOP = list 1..4 of 0..0FFFF, string of 12

Parameters

ADDRESS_PREFIX (AP)

Specifies the address prefix of non-CDCNET NSAP addresses for which the boundary system is being defined. The elements of each string defining an address prefix must be hexadecimal digits. An address prefix must include an even number of hexadecimal digits.

NEXT_HOP (NH)

Specifies information for up to four boundary systems, any of which can be used to reach the non-CDCNET systems whose NSAP addresses are identified via the AP parameter.

Information about each boundary system is specified by a subnet ID and SNPA address pair. The subnet ID identifies a CDCNET subnet directly connected to the boundary system. The SNPA address is the layer 1 address of the boundary system on this subnet. It is same as the system ID of the boundary system.

The subnet ID is specified as an integer with a value between 0 and 0ffff.

The SNPA address is specified as a string containing 12 hexadecimal digits.

Responses The following header precedes the added address prefixes.

The NonCDCNET NSAP address prefixes are added as follows (see example):

The following responses are error messages.

--ERROR-- The NonCDCNET NSAP address prefixes are already defined as follows (see examples for the format):

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains non-hexadecimal characters.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains an odd number of digits.

--ERROR-- SNPA <next_hop SNPA> contains non-hexadecimal characters.

--ERROR-- This command cannot be executed in a system whose routing function is END SYSTEM.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
add_non_cdc_address_prefix ap='00470023' nh = ((15,'0800271000EE'),
(1,'080027100123'))

The NonCDCNET NSAP address prefixes are added as follows:

NonCDCNET NSAP address prefix      Subnet ID    Boundary System SNPA
00470023                             000F           0800271000EE
                                     0001           080027100123


add_non_cdc_address_prefix ap='004700FG' nh=((15,'00EEFF3324EE'))

  --ERROR-- The NonCDCNET NSAP address prefix 004700FG contains
non-hexadecimal characters.


add_non_cdc_address_prefix ap='004700FG' nh=((15,'00EEFG3324EE'))

  --ERROR-- SNPA 00EEFG3324EE contains non-hexadecimal characters.
The NonCDCNET NSAP address prefix 004700FG contains
non-hexadecimal characters.
```

ADD_NON_CDC_ROUTING_ENTRY (ADDNCRE) DI Configuration Procedure/NETOU Command

Purpose

Configures the information needed to route traffic to a non-Control Data system. This command is executed in a CDCNET boundary system. A CDCNET boundary system is the last CDCNET system on the path to a non-Control Data destination. It is connected to the same medium to which a non-Control Data system is also connected. This non-Control Data system is either the final destination or the first non-Control Data next hop on the path to it.

Routing entries defined by the ADD_NON_CDC_ROUTING_ENTRY command are automatically deleted if the next hop trunk associated with these entries is cancelled using one of the CANCEL_XXX_TRUNK commands.

NOTE

CDCNET supports routing to non-Control Data systems only through Ethernet and HDLC trunks for CLNS traffic and through Ethernet and X.25 trunks for CONS traffic.

Format

ADD_NON_CDC_ROUTING_ENTRY
ADDRESS_PREFIX = list 1..10 of string 2..40
NEXT_HOP_TRUNK = name
NEXT_HOP_SNPA = string 1..15
DISTRIBUTE = boolean

Purpose

ADDRESS_PREFIX (AP)

Specifies the address prefix of NSAP addresses for which the next hop is being defined. The elements of each string must be an even number of hexadecimal digits.

NEXT_HOP_TRUNK (NHT)

Specifies the logical name of the trunk which connects the local system to the next hop non-Control Data system. It should be same as the name used in the DEFINE_XXX_TRUNK command used to define the trunk.

NEXT_HOP_SNPA (NHS)

Specifies the SNPA address of the next hop system on the next hop trunk. Its value is specified as a string. The size of the string depends on the type of the medium associated with the next hop trunk.

For Ethernet, the SNPA address is the Ethernet station ID and is specified as a string containing 12 hexadecimal digits.

For X.25 PDN, the SNPA address is the DTE address and is specified as a string containing 1 to 15 decimal digits.

This parameter is not required if the next hop trunk is an HDLC trunk. In this case, if a value is specified for this parameter, it is ignored.

Normally, this parameter is required if the next hop trunk is an Ethernet trunk. However, a value for this parameter may not be specified if the non-Control Data destination systems identified by the ADDRESS_PREFIX parameter, support the ES-IS routing protocol and are directly connected to the Ethernet trunk identified by the NEXT_HOP_TRUNK parameter.

A value must be specified for this parameter if the next hop trunk is an X.25 trunk.

DISTRIBUTE (D)

Indicates whether the specified address prefix(es) should be distributed by the CDCNET IS-IS protocol to other CDCNET systems. If the address prefix(es) are distributed, intermediate systems can learn the route to the boundary system dynamically, as opposed to configuring intermediate systems with ADD_NON_CDC_ADDRESS_PREFIX commands. The default is TRUE.

Responses The following header precedes the routing entry display (see example).

The routing entry is added as follows:

The following response appears if address prefixes are already defined (see examples).

--ERROR-- The routing information for the NonCDCNET NSAP address prefixes is already defined as follows:

The following headers indicate an error message.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains non-hexadecimal characters.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains an odd number of digits.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- LLC 2 trunk <trunk_name> may not be used as the next_hop_trunk.

--ERROR-- SNPA <next_hop_snpa> contains non-hexadecimal characters.

--ERROR-- SNPA <next_hop_snpa> for the X.25 trunk <next_hop_trunk> contains non-decimal characters.

--ERROR-- SNPA <next_hop_snpa> for the ETHERNET trunk <next_hop_trunk> must contain 12 digits.

--ERROR-- This command cannot be executed in a system whose routing function is END SYSTEM.

--FATAL-- The maximum number of NonCDCNET NSAP address prefixes is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Remarks In order to reach a non-Control Data system, routing information must be configured by the ADD_NON_CDC_ROUTING_ENTRY command in each boundary system through which the non-Control Data destination can be reached. This routing information is dynamically distributed to other systems through either the standard ES-IS protocol or the CDCNET IS-IS protocol.

When routing information defined by executing both ADD_NON_CDC_ROUTING_ENTRY and ADD_NON_CDC_ADDRESS_PREFIX commands in the same system reaches the same destination, the information defined by the ADD_NON_CDC_ROUTING_ENTRY command takes precedence. The static routing information received by routing PDUs has lower precedence than that statically configured in a given system.

For CLNS traffic, dynamic routing is attempted first; if it fails, static routing is attempted. For CONS traffic, static routing is attempted first; if it fails, dynamic routing is attempted.

The maximum number of non-Control Data address prefixes which may be configured in a given boundary system is 71-N. N represents the number of directly connected subnets. A minimum value of 10 is assumed for N. Therefore, no more than 61 non-Control Data address prefixes can be configured in a given CDCNET system. However, if the actual number of directly connected subnets is greater than 10, then the actual number is used for N resulting in a lower upper limit for the maximum number of non-Control Data address prefixes.

Examples

```
add_non_cdc_routing_entry ap= ('00470023', '390840')..
nht = ethernet_trunk_one nhs = '0800271000EE'

 The routing entry is added as follows:

 NSAP address prefixes        00470023
                              390840
 Next hop trunk               ethernet_trunk_one
 Next hop SNPA address        0800271000EE
 Distribute                   TRUE


add_non_cdc_routing_entry ap= '004700EE' nht = x25_trunk_1
nhs = '293117234053' d=FALSE

 The routing entry is added as follows:

 NSAP address prefixes        004700EE
 Next hop trunk               x25_trunk_1
 Next hop SNPA address        293117234053
 Distribute                   FALSE


add_non_cdc_routing_entry ap='004700FG' nht= $ESCI01 nhs = '08002530413F'

--ERROR-- The NonCDCNET NSAP address prefix 004700FG contains
non-hexadecimal characters.


add_non_cdc_routing_entry ap='004700F' nht= $ESCI01 nhs = '08002530413F'

--ERROR-- The NonCDCNET NSAP address prefix 004700F contains
an odd number of digits.
```

ADD_NP_GW_OUTCALL (ADDNGO) (NOS only) DI Configuration Procedure/NETOU Command

Purpose Creates a NOS Network Products (NP) gateway outcall definition. Outcall is from the perspective of the CDCNET network; that is, the call goes out of CDCNET. Outcall information generates the proper call request into the NOS host. An NP gateway outcall definition consists of an NP application name and the corresponding CDCNET title by which the gateway is known throughout CDCNET. Titles are site-definable and conform to CDCNET title conventions. Example titles are PTFS$M80 and QTFS$LID.

See also the DEFINE_NP_GW command described in this manual, which explains types of CDCNET titles.

Format

ADD_NP_GW_OUTCALL
TITLE = any
NP_APPLICATION_NAME = string 1..7
GATEWAY_NAME = name

Parameters

TITLE (T)

Title CDNA applications use to access a NOS application through this gateway. The title supports calls from CDNA systems to the NOS host. The title must be unique to the gateway (including titles specified on DEFINE_NP_GW). The title can be of type name (up to 31 characters), or type string (up to 255 characters, within apostrophes).

NP_APPLICATION_NAME (NAN)

NOS application name accessed when a CDNA application (or gateway) initiates a call connection with the corresponding CDNA title.

GATEWAY_NAME (GN)

NP gateway name (see DEFINE_NP_GW command described in this manual) providing access to the NOS application. The gateway must be previously defined. If this command is specified in a configuration file, default is the previously defined NP 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.
```

ADD_OUTGOING_SERVICE_ACCESS (ADDOSA)
DI Configuration/NETOU Command

Purpose Adds the titles of servers that may be accessed going out via the network solution.

Format

ADD_OUTGOING_SERVICE_ACCESS
TITLE = list 1..15 of name or string 1..255
NETWORK_NAME = name

Parameters

TITLE (T)

Title of a server that may be accessed going out via the network solution. The title may contain wildcard characters to specify a set of allowed titles (see the CDCNET Network Operations and Analysis manual for information on wildcard characters).

Title is case-sensitive when entered as a string. For example, if the command defining a service specifies the service title in uppercase, it must be specified in uppercase for the ADDOSA command or it is seen as a different title.

For more information on the use of service titles, see the CDCNET Configuration manual.

NETWORK_NAME (NN)

Identifies the name of a currently defined network within this system. This is the network solution upon which the offered services are being defined. Defaults to the previously defined network when on the configuration file. This parameter is required when the command is entered by the operator.

Responses

Outgoing service access added to network <network_name>.

--WARNING-- Outgoing service access added to network <network_name>.
Title <title> already added to network.
{Line repeats for each title already added to network}

--ERROR-- A network is not defined.

--ERROR-- Network <network_name> is not defined.

--ERROR-- Title <title> may not appear more than once in the list of specified titles.

--ERROR-- Expecting NAME or STRING, found <xxx>.

--FATAL-- Not enough memory is currently available for required table space.

Examples In this example, an HDLC trunk links two separate sites (figure 3-4). Assume that service access between the two sites must be limited to the following:

- QTF/PTF file transfers in either direction.
- Site 1 has unrestricted interactive service access to site 2.
- Site 2 interactive service access to site 1 is limited to Host_1.

Add the following to the configuration file for NDI_21.

```
define_hdlc_net..
   trunk_name=hdlc_trunk_1,network_name=hdlc_net_1,..
   network_id=3..
   incoming_service_access=(file_transfer,interactive)..
   outgoing_service_access=(file_transfer)
add_outgoing_service_access title=host_1
```

The DEFINE_HDLC_NET command prevents interactive attempts from going out on the network, thus eliminating extra traffic.

The ADD_OUTGOING_SERVICE_ACCESS command allows access to HOST_1.

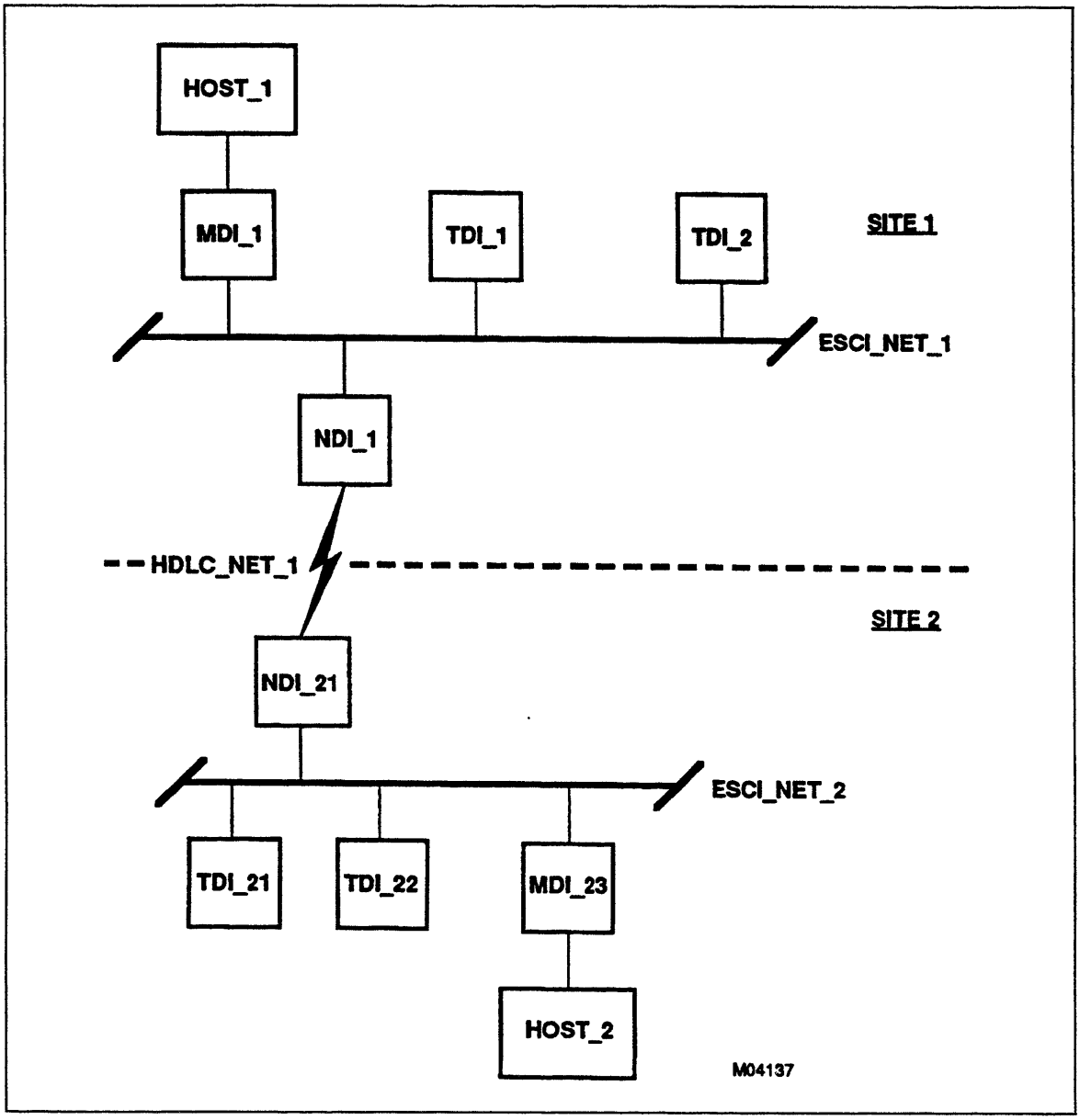


Figure 3-4. ADDOSA Example

ADD_RIP_SERVICE (ADDRS)
DI Configuration Command/NETOU Command

Purpose

Adds Routing Information Protocol (RIP) support on directly connected IP networks/subnets. When RIP is configured, CDCNET receives and/or sends RIP datagrams over the network. The RIP datagrams received are used to update the routing information in the DI for routing IP datagrams. When processing received datagrams, RIP does not change existing route table entries for statically configured routes (for example, owner = IPSR) or for routes owned by EGP or ICMP.

The RIP datagrams sent contain routing information that other RIP hosts on the network can use to route IP datagrams through the DI. RIP datagrams are sent when requested by a RIP peer, either periodically (every 30 seconds) on all directly connected networks with RIP configured, or when an update received from a gateway on one network is relayed to other directly connected networks.

RIP processes default routes (IP address = 0.0.0.0). RIP does not change a statically configured default route. RIP sends a known default route, unless excluded using the EXCLUDED_ROUTES parameter.

The User Data Protocol (UDP) must be configured by the DEFINE_UDP_INTERFACE command before the RIP service is configured. IP broadcast must also be enabled on the IP network for which RIP service is added. Use the ENABLE_IP_BROADCAST parameter on the DEFINE_IP_NET command.

Format

ADD_RIP_SERVICE
IP_NETWORK = list 1..4 of 0..255
OPERATING_MODE = keyword value
TRUNK_COST = integer 1..15
TRUSTED_NEIGHBORS = list 1..5 of list 1..4 of 0..255
EXCLUDED_ROUTES = list 1..20 of list 1..4 of 0..255
INCLUDED_ROUTES = list 1..20 of list 1..4 of 0..255
DISABLE_ARP_ROUTING = boolean
POISONED_REVERSE = boolean

Parameters

IP_NETWORK (IN)

Specifies the IP network number of the directly connected network for which the RIP protocol is to be used. This network must have been defined by a DEFINE_IP_NET command,

OPERATING_MODE (OM)

Specifies how RIP is used on this directly connected network. This parameter deals with the processing of RIP update messages and not the processing of RIP request messages. CDCNET responds to all RIP request messages. The default is BOTH.

The possible keyword values are:

Keyword Value	Description
BOTH	RIP update messages should be sent on this network. Received RIP update messages should be processed.
RECEIVE	RIP update messages should be sent on this network only when the update is requested from a peer. Received RIP update messages should be processed.
SEND	RIP update messages should be sent on this network. Received RIP update messages should not be processed.

TRUNK_COST (TC)

Specifies the RIP cost for this directly connected network. A RIP update message received from a peer contains a cost value for each network reported. The reported cost is incremented by this parameter value prior to creating or updating the IP routing table information for the reported network. The cost, which is stored in IP routing table entries created by RIP, is the same as the HOP_COUNT value for routing table entries statically configured by the DEFINE_IP_NET command. With RIP, a network whose cost equals 16 (or greater) is considered unreachable. This parameter is meaningful only if the OPERATING_MODE parameter is BOTH or RECEIVE. The default is 1. You can use a value other than 1 to show preference between directly connected networks that both provide routes to the same remote network.

TRUSTED_NEIGHBORS (TN)

Specifies a list of IP addresses for neighbor hosts from which incoming RIP updates are considered valid. Only updates from these neighbors are candidates for inclusion in the local IP routing tables. This parameter is meaningful only if the OPERATING_MODE parameter is BOTH or RECEIVE. The default is ALL.

EXCLUDED_ROUTES (ER)

Specifies a list of IP network numbers not to be included in RIP update messages sent on this directly connected network. This parameter is meaningful only if the OPERATING_MODE parameter is BOTH or SEND. The default is NONE.

INCLUDED_ROUTES (IR)

Specifies a list of IP network numbers to be included in RIP update messages sent on this directly connected network. This parameter is meaningful only if the OPERATING_MODE parameter is BOTH or SEND. The default is ALL.

DISABLE_ARP_ROUTING (DAR)

Specifies whether ARP proxy routing should be disabled or remain enabled on the network. A value of TRUE indicates ARP proxy routing is to be disabled; FALSE indicates it should remain enabled. The default is TRUE. This parameter should be TRUE unless situations call for both RIP and ARP proxy routing to be enabled.

POISONED_REVERSE (PR)

Specifies whether to use split horizon with poisoned reverse routing or simple split horizon routing when issuing network information back to the network that the data was received from. A value of TRUE indicates split horizon with poisoned reverse when network information is issued with an infinite metric. A value of FALSE indicates simple split horizon when network information is not issued on these networks. The default is TRUE.

Responses

RIP service is added for IP network/subnet <ip_network>.

--ERROR-- UDP is not defined.

--ERROR-- RIP service is already defined for IP network/subnet <ip_network>.

--ERROR-- IP broadcast is not enabled for the specified IP network.

--ERROR-- IP network <ip_network> is invalid.

--ERROR-- IP network <ip_network> is not defined.

--ERROR-- IP subnet <ip_network> is invalid.

--ERROR-- IP subnet <ip_network> is not defined.

--ERROR-- IP network <ip_network> is not directly connected.

--ERROR-- IP address <trusted_neighbor> is not a host on this directly connected network.

--ERROR-- IP host address <excluded_routes> cannot be specified as an excluded route.

--ERROR-- IP host address <included_routes> cannot be specified as an included route.

--FATAL-- Not enough memory is currently available for required table space.

Remarks

RIP service does not recognize RIP update messages for a default route. The user must configure the DEFAULT ROUTE entries in the DI.

ADD_SNMP_COMMUNITY (ADDSC)
DI Configuration Command/NETOU Command

Purpose Adds support for an SNMP community.

Format

ADD_SNMP_COMMUNITY
COMMUNITY = string 1..32
IP_ADDRESS = list 1..4 of 0..255
SEND_TRAPS = boolean
ACCESS_MODE = keyword value

Parameters

COMMUNITY (C)

Specifies the community name for which to add support. The SNMP agent uses the list of supported community names upon receipt of a PDU in its trivial authentication scheme.

IP_ADDRESS (IA)

Specifies the IP address associated with the input community name. If there are no IP addresses associated with a particular community name, all IP addresses are accepted, but no traps are sent to those IP addresses. If there are IP addresses associated with a community name, only PDUs from the IP addresses with that community name are deemed authentic. Traps are sent only to IP addresses tconfigured with the SEND_TRAPS parameter set to TRUE. The SNMP Agent allows PDUs from any IP address that has a community name of *public* and READ_ONLY access.

SEND_TRAPS (ST)

Specifies whether the input IP_ADDRESS/COMMUNITY pair should be sent SNMP Agent Traps. If TRUE, the SNMP Agent generates and sends a cold start trap to the defined IP address and community.

ACCESS_MODE (AM)

Specifies the access mode the IP_ADDRESS/COMMUNITY pair is allowed. If READ-ONLY, you cannot set any variables. If READ_WRITE, you can set nonsensitive variables in MIB-II to READ_WRITE access. You cannot set sensitive variables, such as variables in the IP Routing Table. The default is READ-ONLY.

Responses

The SNMP community '<community>' has been added.

IP address <ip address> for SNMP community '<community>' has been added.

--ERROR-- The SNMP Agent is not defined.

--ERROR-- The SNMP community '<community>' has already been defined.

--ERROR-- IP address <ip_address> for SNMP community '<community>' has already been defined.

--ERROR-- IP_ADDRESS is a required parameter if SEND_TRAPS is specified as TRUE.

--FATAL-- Not enough memory is currently available for required table space.

ADD_TCPIP_ACCESS (ADDTA) (NOS/VE Only)
DI Configuration Command/NETOU Command

Purpose Adds TCP/IP services to a previously defined NOS/VE interface which allows access to TCP/IP applications on a NOS/VE host.

NOTE

For NOS applications, use the DEFINE_TCPIP_GW command described in this manual.

Format

ADD_TCPIP_ACCESS
IP_ADDRESS = list 4 of 0..255
INTERFACE_NAME = name
MAX_LISTEN_PORTS = integer 1..10

Parameters

IP_ADDRESS (IA)

Specifies the IP address of the NOS/VE CYBER host for which this access provides service. This IP address must have been previously defined by a DEFINE_IP_HOST command. The format of this parameter is similar to the decimal octet convention used by the TCP/IP community, except that the periods are replaced by commas, and the list is enclosed in parentheses. For example, the IP address 128.2.53.7 would be represented as (128,2,53,7).

INTERFACE_NAME (IN)

Specifies the logical name of the NOS/VE interface to be used for TCP/IP access. Before executing this command, the NOS/VE interface must be configured, either explicitly by using the DEFINE_VE_INTERFACE command or implicitly by booting the DI system over the channel running the NOS/VE interface.

This parameter is required if more than one MCI board exists in the DI.

MAX_LISTEN_PORTS (MLP)

Specifies the maximum number of listen sockets allowed per port when an application from NOS/VE opens a TCP SAP. The lesser of two values, the one specified by the TCP/IP application on the open SAP call and the other the value of this parameter, is used. The default is 5.

Responses

TCP/IP access is added for VE interface <interface_name>.

--WARNING-- TCP/IP access is already added for VE interface <interface_name>.

--ERROR-- No VE interface is defined.

--ERROR-- VE interface <interface_name> is not defined.

--ERROR-- Interface <interface_name> is not a VE channel interface.

--ERROR-- IP address <ip_address> is invalid.

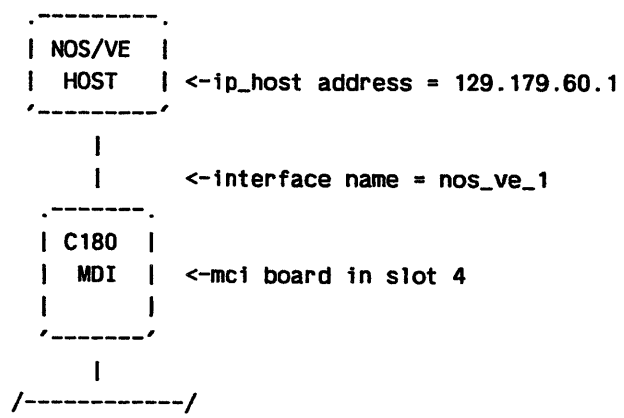
--ERROR-- IP host address <ip_address> is not defined.

--ERROR-- IP host with address <ip_address> must be of host type LOCAL or CDC_HOST.

--ERROR-- The Device Interface contains more than one CHANNEL board the interface_name must be specified.

--FATAL-- Not enough memory is currently available for required table space.

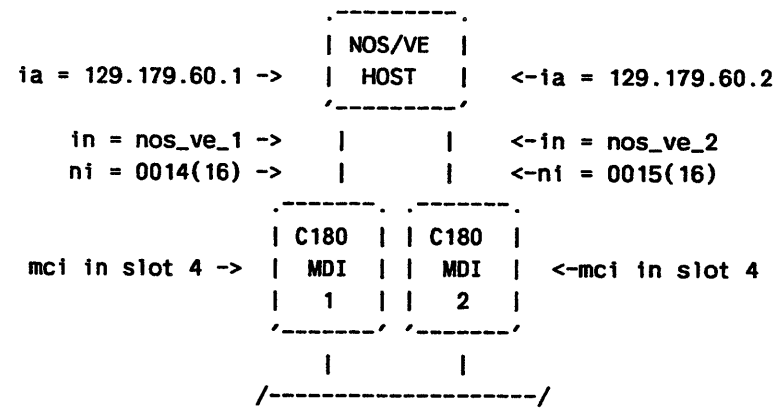
Examples The first example uses a host with only one channel to a DI.



The following supporting configuration commands are used for the MDI.

```
define_ve_interface interface_slot=4 interface_name=nos_ve_1
define_ip_net ip_network=(129,179,60) trunk_name=nos_ve_1 ..
       subnet_mask_size=22
define_tcp_interface
define_udp_interface
define_ip_host ip_address=(129,179,60,1) host_type=local
add_tcpip_access ip_address=(129,179,60,1) ..
       interface_name=nos_ve_1
```

The second example uses a host with two channels to two DIs.



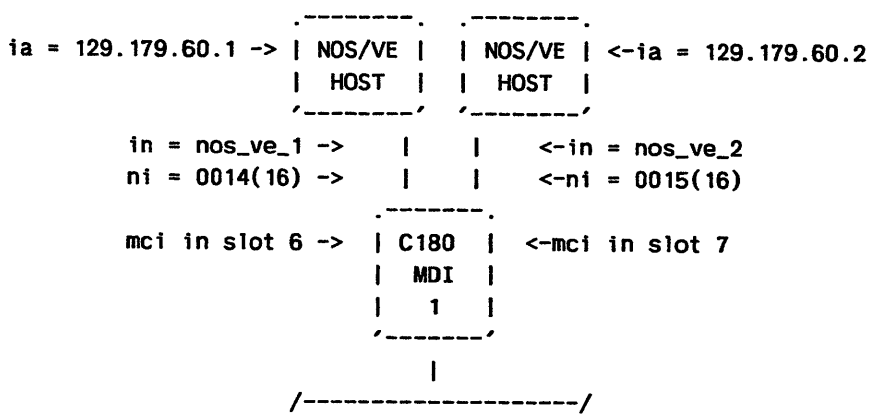
The following supporting configuration commands are used for MDI 1.

```
defvi interface_slot=4 interface_name=nos_ve_1 ..
       network_id=0014(16)
define_ip_net ip_network=(129,179,60) trunk_name=nos_ve_1 ..
       subnet_mask_size=22
define_tcp_interface
define_udp_interface
define_ip_host ip_address=(129,179,60,1) host_type=local
add_tcpip_access ip_address=(129,179,60,1) ..
       interface_name=nos_ve_1
```

The following supporting configuration commands are used for MDI 2.

```
defvi interface_slot=4 interface_name=nos_ve_2 ..
      network_id=0015(16)
define_ip_net ip_network=(129,179,60) trunk_name=nos_ve_2 ..
       subnet_mask_size=22
define_tcp_interface
define_udp_interface
define_ip_host ip_address=(129,179,60,2) host_type=local
add_tcpip_access ip_address=(129,179,60,2) ..
       interface_name=nos_ve_2
```

The third example shows two hosts with one channel each to the same DI.



The following supporting configuration commands are used for the MDI.

```
defvi interface_slot=6 interface_name=nos_ve_1 ..
       network_id=0014(16)
defvi interface_slot=7 interface_name=nos_ve_2 ..
       network_id=0015(16)
define_ip_net ip_network=(129,179,60) trunk_name=nos_ve_1 ..
       subnet_mask_size=22
define_tcp_interface
define_udp_interface
define_ip_host ip_address=(129,179,60,1) host_type=local
define_ip_host ip_address=(129,179,60,2) host_type=cdc_host
add_tcpip_access ip_address=(129,179,60,1) ..
       interface_name=nos_ve_1
add_tcpip_access ip_address=(129,179,60,2) ..
       interface_name=nos_ve_2
```

ADD_TELNET_SERVER (ADDTS)
DI Configuration Procedure/NETOU Command

Purpose

Configures an instance of the TELNET Server into the DI. Each instance of a TELNET Server is associated with a unique IP address and PORT number. Multiple TELNET Servers can be configured into a DI.

The command DEFINE_TIP TN=TELNETTIP is used to configure the TELNET TIP into the DI and must be executed before any ADD_TELNET_SERVER commands. The DEFINE_TIP command is also used to establish the default values for some of the parameters of the ADD_TELNET_SERVER command.

Format

ADD_TELNET_SERVER
SERVER_NAME = name
IP_ADDRESS = list 4 of 0..255
TCP_PORT_NUMBER = 0..65535
PROTOCOL = TELNET or *NONE*
TERMINAL_DEFINITION_PROCEDURE = name or *none*
TERMINAL_USER_PROCEDURE = name or *none*
PROCEDURE_FILE_OPTION = IP_ADDRESS_CONCATENATION or *none*
CONNECTION_CONNECT_TIMEOUT = 20..1000 or *INFINITE*
CONNECTION_DISCONNECT_TIMEOUT = 0..1000 or *INFINITE*
USER_CONNECTION_LIMIT = 1..16
INITIAL_OPTION_NEGOTIATION = list 1..4 of keyword
VALIDATE_USERS = boolean
TCP_TIMEOUT = 0..65535 or *INFINITE*

Parameters

SERVER_NAME (SN)

Logical name of the specified TELNET Server. This name is used in subsequent commands that reference this TELNET Server. It also specifies the line name for other terminal support commands such as WRITM.

IP_ADDRESS (IA)

IP address associated with this TELNET Server. 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, IP address 128.2.53.7 is specified as (128,2,53,7).

TCP_PORT_NUMBER (TPN)

TCP port number associated with this TELNET Server. Default is 23 which is the standard value.

PROTOCOL (P)

Specifies the protocol used on the TCP connection. TELNET specifies that TELNET protocol is used. For incoming terminal connections, select TELNET. If *lined* is to be used on a UNIX system to connect to this server for access to a CDCNET printer, select NONE. NONE specifies that no protocol above TCP is used. Default is TELNET.

TERMINAL_DEFINITION_PROCEDURE (TDP)

Specifies the TDP automatically executed for all users when they first access this TELNET Server. NONE specifies that no TDP is to be automatically executed. The default value for this parameter is derived from the value of the DEFINE_TIP command parameter with the same name.

TERMINAL_USER_PROCEDURE (TUP)

Specifies the TUP automatically executed for all users when they first access this TELNET Server. NONE specifies that no TUP is to be automatically executed. This parameter can be used only when the TDP parameter has a value of NONE. The default value for this parameter is derived from the value of the DEFINE_TIP command parameter with the same name.

PROCEDURE_FILE_OPTION (PFO)

Specifies one of two options for qualifying the procedure name specified by either the TDP or TUP parameter. Specifying IP_ADDRESS_CONCATENATION causes the remote IP address identified within the incoming connection to be concatenated with the name specified by either the TDP or TUP in effect for this Server. If neither a TDP nor TUP is in effect for this Server, this parameter is ignored. NONE selects no concatenation. Default is NONE.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines how much time the TELNET user has to create the first $input/$output connection. If no connection is established within that time, the TELNET connection is terminated. It may be necessary to increase this value if the VALIDATE_USERS parameter is TRUE. INFINITE indicates an infinite time. The timer has a precision of +/- 2 seconds. Default is 120.

CONNECTION_DISCONNECT_TIMEOUT (CDT)

Defines how much time the TELNET 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 user's connection terminates. INFINITE indicates an infinite time. Default is 120.

USER_CONNECTION_LIMIT (UCL)

Defines the maximum number of interactive $input/$output connections allowed at any one time by a TELNET user. Default is 4.

INITIAL_OPTION_NEGOTIATION (ION)

Specifies the set of TELNET options to be negotiated (transmitted) by this TELNET Server during the initial startup of all TELNET connections. You may specify any subset of the keywords, WILL_SGA, WILL_ECHO, DO_SGA, and DO_TTYPE or the single keyword NONE. A list containing the keyword NONE is treated as if NONE were the only keyword specified. The keyword NONE specifies that no TELNET option negotiation is to be initiated by this TELNET Server. This parameter is ignored when the PROTOCOL parameter has a value of NONE. Default is (WILL_SGA, WILL_ECHO).

Keyword Value	Description
DO_SGA	The server asks the client to suppress the Go Ahead option.
DO_TTYPE	The server asks the client to negotiate terminal type. The received terminal type is used as the TERMINAL_MODEL terminal attribute.
WILL_ECHO	The server offers to do character echoing.
WILL_SGA	The server offers to suppress the Go Ahead option.
NONE	The server negotiates no TELNET options.

VALIDATE_USERS (VU)

Specifies if the TELNET users accessing this TELNET Server are validated when user validation is enabled by the DEFINE_USER_VALIDATION command. If VU is TRUE, users are validated when validation is enabled. The default value for this parameter is derived from the value of the DEFINE_TIP command parameter with the same name.

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. Entering a small value (less than a few seconds) may cause frequent and unnecessary loss of service during periods of network congestion. Entering a large value may cause users to wait a long time after the host or network has failed. If you enter INFINITE, the connection will not time out. The default value is 300 seconds.

The TELNET TIP uses this parameter to determine how often to send TELNET NOP commands on inactive connections, ensuring the connection aborts if the remote host is no longer available.

Responses

TELNET server <server_name> is added.

--ERROR-- The TELNET TIP must be defined prior to adding TELNET servers.

--ERROR-- TELNET server <server_name> is already added.

--ERROR-- IP address <ip_address> is not defined.

--ERROR-- IP address and port <ip_address.port> is already in use.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
add_telnet_server server_name=arhnet ip_address=(128,2,53,7)

TELNET server ARHNET is added.
```

ADD_X25_GW_INCALL (ADDXGI)
DI Configuration Procedure/NETOU Command

Purpose Defines calling DTE addresses that can access CDCNET and titles to which they may connect. A title can be a NOS/VE application title, a Network Products (NP) gateway title, or an X.25 A-to-A gateway title. The first time a DTE address is specified on ADDXGI, an entry is made in a list of incall validations the gateway performs. If a DTE address is specified that has been specified previously, an error results.

Format

ADD_X25_GW_INCALL
REMOTE_DTE_ADDRESSES = (list 1..31 of string 1..15) or OTHER
GATEWAY_NAME = name
ALLOWED_TITLES = list 1..31 of (list 1..2 of (string or *name* or *keyword value)* or *boolean)*

Parameters

REMOTE_DTE_ADDRESSES (RDA)

One or more allowed DTE addresses. When the X.25 call is received, and the calling DTE address is not in the list of DTE addresses of any ADDXGI command, the call is cleared. OTHER indicates any DTE address not specifically listed in any other INCALL command for the gateway.

GATEWAY_NAME (GN)

Name of the X.25 A-to-A gateway receiving incoming calls to be validated. The gateway must be previously defined. This parameter value depends on how ADDXGI executes.

If ADDXGI executes in a configuration procedure, this parameter is **optional**, and the default network name is the previously defined X.25 gateway.

If ADDXGI is entered by a network operator, this parameter is **required**.

ALLOWED_TITLES (AT)

List of titles the caller can access, and, for each title, whether the caller can use reverse charging. Titles can contain wildcard characters to specify a set of allowed titles. The following rules apply to titles and wildcard characters specified on ADDXGI:

- If ALL, the * wildcard character, or no titles are specified, calls to any title are accepted.
- If a title or pattern is specified more than once, an error results.
- If an incoming title matches more than one allowed title pattern, the first pattern that is matched is used.

The following keyword values on ALLOWED_TITLES specify whether or not the caller can use reverse charging:

Keyword Value	Description
YES	Service can be accessed with reverse charging (reverse charging is accepted).
NO	Calls with reverse charging are not accepted.

Default is NO.

Responses

X.25 incall added to gateway <gateway_name>.

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <name> is not defined.

--ERROR-- Remote_dte_address can not include <string>. A remote_dte_address may include only digits 0 thru 9.

--ERROR-- Address <string> is already specified as a remote_dte_address.

--ERROR-- Title <string> is already specified as an allowed_title.

--FATAL-- Not enough memory is available for required table space.

Examples

The following example specifies that gateway X25GW accepts all incoming calls accepted with reverse charging requested.

```
add_x25_gw_incall gateway_name=X25GW,..
  remote_dte_address=OTHER, allowed_titles = (ALL,YES)
```

The following example specifies that incoming calls from DTE addresses 408123, 408456, and 612789 to gateway X25GW can connect to any title beginning with 'PTFS$'. Calls with reverse charging specified are not allowed.

```
add_x25_gw_incall gateway_name=X25GW,..
  remote_dte_addresses=('408123','408456','612789'),..
  allowed_titles='PTFS$*'
```

The following example specifies that gateway X25GW accepts only incoming calls from DTE addresses 4081234, 4083456, and 6126789 and requesting titles beginning with PTFS$ and QTFS$ using reverse charging. Calls from all other DTE addresses can connect to PTFS$PUBLIC and QTFS$PUBLIC, without reverse charging.

```
add_x25_gw_incall gateway_name=X25GW,..
  remote_dte_addresses=('4081234', '4083456','6126789'),..
  allowed_titles=(('PTFS$*',YES), ('QTFS$*',YES))

add_x25_gw_incall gateway_name=X25GW,..
  remote_dte_addresses=OTHER,..
  allowed_titles=(('PTFS$PUBLIC'),('QTFS$PUBLIC'))
```

ADD_X25_GW_OUTCALL (ADDXGO)
DI Configuration Procedure/NETOU Command

Purpose

Defines a gateway outcall definition. Outcall is from the perspective of the CDCNET network; that is, the call goes out of CDCNET. Outcall information generates the proper call request into the X.25 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. See also the DEFINE_X25_GW command description in this manual.

Format

ADD_X25_GW_OUTCALL
TITLE = string 1..255
REMOTE_DTE_ADDRESS = string 1..15
PROTOCOL_ID = 0..255
GATEWAY_NAME = name
LOCAL_DTE_ADDRESS = string 1..15
FACILITIES = string 1..63
USER_DATA = string 2..248

Parameters

TITLE (T)

Title CDNA applications use to access a particular remote application through this gateway. The title supports calls from CDNA systems to remote systems accessed through the X.25 network.

REMOTE_DTE_ADDRESS (RDA)

X.25 address of the destination X.25 system, specified as a string of digits 0 through 9.

PROTOCOL_ID (PI)

Protocol identifier required by the destination X.25 system. This value is placed in the first octet of the protocol identifier field within the call user data in a call request. Octets 2 through the protocol identifier field are set to 0 by the gateway.

GATEWAY_NAME (GN)

Name of the X.25 gateway providing access to the remote application. The gateway must be previously defined. If this command is specified in a configuration file, default 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 (LDA)

X.25 address of a local X.25 trunk, specified as a string of digits with values from 0 through 9. 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. Default is that the X.25 Packet Level selects a trunk to make the call request.

FACILITIES (F)

Facilities options as defined by the X.25 CCITT standard. For information on X.25 facilities options, see CCITT Recommendation X.25. Specify this parameter as an even-numbered string of hexadecimal digits.

USER_DATA (UD)

An even-numbered string of hexadecimal digits. This value is put at the beginning of any "real" user data from the session indication, and the concatenated string is then put 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

The following detailed example shows how the ADDXGO command builds a call request.

```
add_x25_gw_outcall ..
     title = ptfs$vw ..
     gateway_name = x25_gateway ..
     remote_dte_address='311061200046' ..
     protocol_id=0c2(16) ..
     facilities='420707430202'
     user_data='50544653245657'
```

This example sends the following X.25 call request when an application calls (connects) to title $PTFS$VW:

```
0B 0C 31 10 61 20 00 46 06 42 07 07 43 02 02 ..
     C2 00 00 00 50 54 46 53 24 56 57
```

In a binary format, the call request looks like figure 3-5:

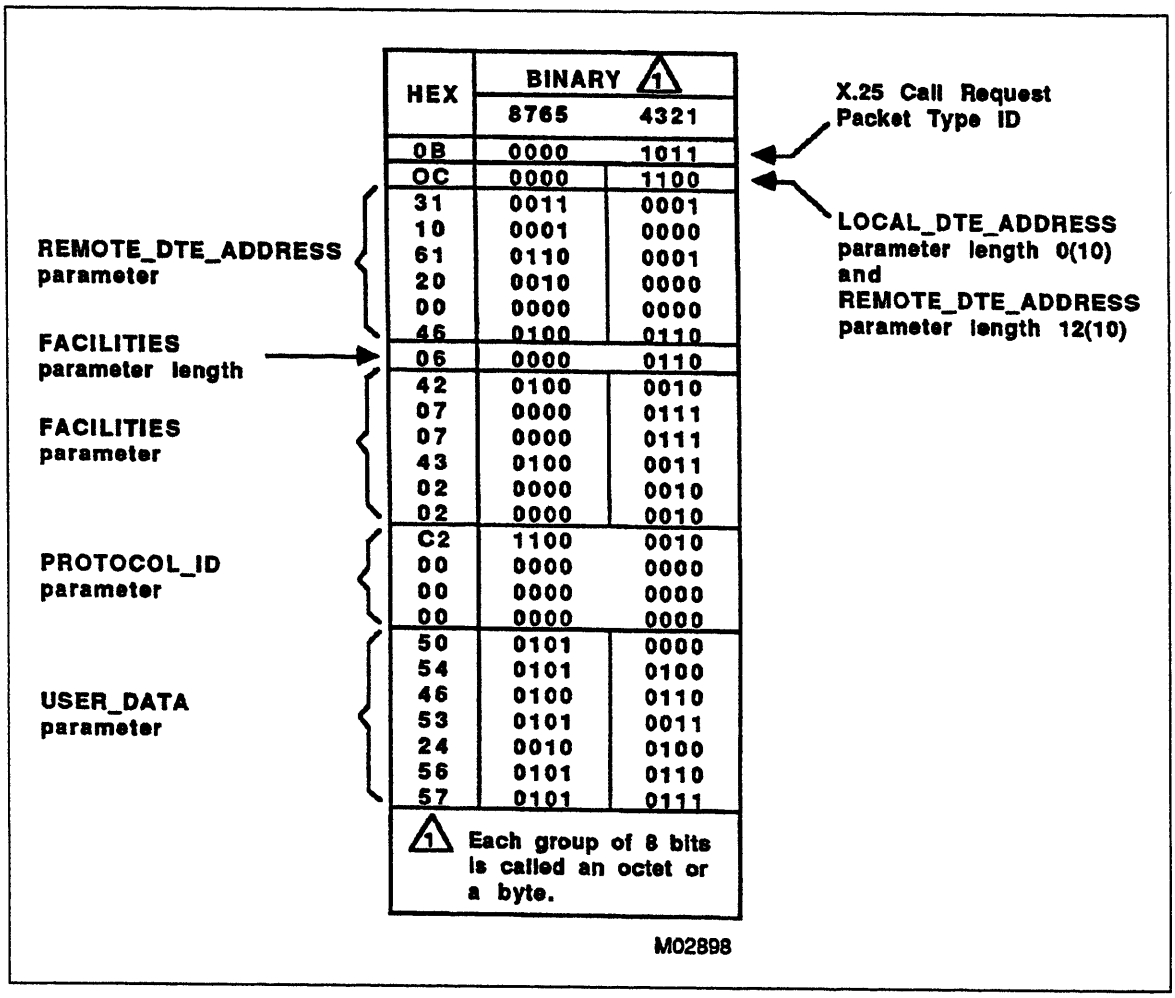


Figure 3-5. X.25 Call Request Generated by ADDXGO Command

If no FACILITIES parameter is specified, the facilities length octet is zero:

```
0B 0C 31 10 61 20 00 46 00 C2 00 00 00 50 54 46 53 24 56 57
```

If no USER_DATA parameter is specified, no call data is included:

```
0B 0C 31 10 61 20 00 46 00 C2 00 00 00
```

If a LOCAL_DTE_ADDRESS parameter is specified, the calling DTE address is included:

```
0B CC 31 10 61 20 00 46 31 10 40 80 01 46 06 42 07 07 43 02 02 ..
   C2 00 00 00 50 54 46 53 24 56 57
```

ADD_X25_GW_PVC (ADDXGP)
DI Configuration Procedure/NETOU Command

Purpose

Defines PVCs for the given gateway. The PVCs can be for either incoming calls or outgoing calls. When the X.25 gateway receives an outgoing call (for example, an application call request), it determines the title based on the user data supplied. If this title is associated with a PVC, it establishes a PVC connection (for example, an X.25 reset request is sent with CAUSE field set to REMOTE DTE OPERATIONAL) and discards any additional user data. The application is responsible for initiating the connection to the CDCNET DI. Once the application has established a connection, if a reset is received with the CAUSE field set to OUT OF ORDER, the connection is terminated. To disconnect the PVC, the application must terminate the connection. This causes a reset to be sent with the CAUSE field set to OUT OF ORDER. Any other resets received with other CAUSE fields create a CDCNET session layer SYNCH request to be sent to the application.

Format

ADD_X25_GW_PVC
TITLE = name
PERMANENT_VIRTUAL_CIRCUIT = 1..4095
GATEWAY_NAME = name
PVC_PACKET_SIZE = 16..4096
PVC_WINDOW_SIZE = 1..127
TRUNK_NAME = name

Parameters

TITLE (T)

Specifies the title that CDCNET applications can use to access a particular remote application through this gateway. The title supports calls from CDCNET systems to remote systems accessed through the X.25 trunk.

PERMANENT_VIRTUAL_CIRCUIT (PVC)

Specifies the permanent virtual circuit with which the given title is associated. The value must be within the range of the PVC_RANGE parameter defined on the DEFINE_X25_INTERFACE command. Its value cannot overlap another defined PVC (for example, X25_ASYNCTIP or a PVC given on a previously defined ADD_X25_GW_PVC).

GATEWAY_NAME (GN)

Specifies the name of the X.25 gateway that provides access to the remote application. The gateway must be previously defined. Default is the previously defined X.25 gateway when included in the configuration file. This parameter is required when entered by an operator.

PVC_PACKET_SIZE (PPS)

Specifies the default packet size to be used for this PVC. The default is the value specified on the DEFAULT_PACKET_SIZE parameter of the associated DEFINE_X25_INTERFACE command. See the description of the DEFINE_X25_INTERFACE command for allowed values.

PVC_WINDOW_SIZE (PWS)

Specifies the default window size to be used for this PVC. The default is the value specified on the DEFAULT_WINDOW_SIZE parameter of the associated DEFINE_X25_INTERFACE command.

TRUNK_NAME (TN)

Specifies the trunk to be used for this PVC. This parameter is required if a list of trunk names was specified on the DEFINE_X25_GW command. If only one trunk name was specified on DEFINE_X25_GW, that is the default trunk and this parameter is not required.

Responses

PVC <pvc> added for title <title>, X.25 gateway <gn>.

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <gateway_name> is not defined.

--ERROR-- X.25 gateway title <title> already defined.

--ERROR-- PVC packet size <pps> is not a valid X.25 packet size.

--ERROR-- PVC window size <pws> is greater than 7 and the DEFINE_X25_INTERFACE PSN parameter is NORMAL for trunk <trunk_name>.

--ERROR-- Permanent Virtual Circuit number <pvc> is not defined on DEFINE_X25_INTERFACE, PR parameter.

--ERROR-- Permanent Virtual Circuit number <pvc> is already assigned by another X.25 user.

--ERROR-- Trunk name <trunk_name> not defined as an X.25 trunk on the X.25 gateway <gateway_name>.

--ERROR-- More than one trunk_name associated with X.25 gateway <gateway_name>. Trunk_name must be specified.

--ERROR-- Permanent Virtual Circuit number <pvc> is already assigned to title <title>, X.25 gateway <gateway_name>.

--ERROR-- X.25 Gateway command failed due to title processor error.

--FATAL-- Not enough memory available for required table space.

Examples

The following example associates a permanent virtual circuit of 100 to title PVC$FRN.

```
add_x25_gw_pvc gateway_name=X25_GW title=PVC$FRN ..
  permanent_virtual_circuit=100

PVC 100 added for title PVC$FRN, X.25 gateway X25_GW
```

ADD_X25_INCALL (ADDXI)
DI Configuration Procedure/NETOU Command

Purpose

Defines a general purpose X.25 incall definition. An incall definition is from the perspective of the CDCNET network. The incall information is used by the X.25 packet-level entity to control X.25 calls coming into the CDCNET network destined for any X.25 user.

A trunk that has no X.25 incall definition does not validate incoming X.25 calls. This is the default situation. Each trunk is handled independently. The means for associating the incoming call with the incall definition is the combination of the remote DTE address (calling DTE address) and the protocol identifier.

In general, the validation configured by ADD_X25_INCALL is independent of the validation configured by any X.25 users. Most X.25 users do not have their own incall validation. However, the X.25 gateway does, as defined by ADD_X25_GW_INCALL. Incoming calls destined for the X.25 gateway are subject to validation by the X.25 packet level as well as the X.25 gateway.

Format

ADD_X25_INCALL
TRUNK_NAME = list 1..32 of name
REMOTE_DTE_ADDRESSES = list 1..31 of string 1..31 or keyword value
DTE_ADDRESS_PROTOCOL_ID = list 1..7 of 0..255 or keyword value
ACCEPT_PDN_CHARGES = boolean

Parameters

TRUNK_NAME (TN)

Specifies the logical names of one or more X.25 trunks for this incall definition. Each trunk must have been defined by a DEFINE_X25_TRUNK or DEFINE_LLC_2_TRUNK command.

REMOTE_DTE_ADDRESSES (RDA)

Specifies one or more allowed calling DTE addresses. A DTE address can contain wild card attributes (to associate the incall definition with multiple DTE addresses) as follows:

[D1-D2] Within a bracketed group, a range of digits can be specified. D1 represents the low end of the range; D2 represents the high end. Any value within the range can be used to match a digit in the position specified in the address. For example, 1234[6-7]59 matches two DTE addresses: 1234659 and 1234759.

[...] Within a bracketed group, any list of single digits can be specified. For example, 1234[179]59 matches three DTE addresses: 1234159, 1234759, and 1234959.

* Any single * can be used to match a string of digits, including the null string. For example, 1234* matches 1234, 12341, 1234562345, and so forth.

? Any single ? can be used to match a digit in the position specified in the address. For example, 1234?59 matches 10 DTE addresses: 1234059, 1234159, 1234259 ... 1234959.

If multiple ADD_X25_INCALL definitions exist that match a given DTE address, the call definition associated with the most specific string is used. For example, 123?56 is more specific than 123* in matching calling DTE address 123456.

The keyword OTHER is used to define an incall definition for any DTE addresses that do not match any other incall definitions for a given trunk. Multiple incall definitions can exist with a REMOTE_DTE_ADDRESSES value of OTHER (as well as any other value) as long as the DTE_ADDRESS_PROTOCOL_ID values do not overlap.

DTE_ADDRESS_PROTOCOL_ID (DAPI)

Specifies one or more protocol identifiers to identify X.25 users associated with the incall definition. The protocol identifier is the first octet of user data in an X.25 call packet. This parameter may be either explicit protocol identifiers or one of the keywords NULL, ANY, or NONE.

Keyword Value	Description
NULL	Value corresponding to incoming calls without user data.
ANY	Any value of the protocol identifier (including NULL) is accepted.
NONE	No value of the protocol identifier (including NULL) is accepted.

The default is ANY.

ACCEPT_PDN_CHARGES (APC)

Specifies how incoming calls are to be handled. If TRUE, a call is accepted with either normal or reverse charging. If FALSE, a call is accepted only with normal charging. The default is FALSE.

Responses

The X.25 incall definition(s) have been added.

--ERROR-- The X.25 incall definition for trunk <trunk_name> is already defined as follows:

Remote DTE Addresses = <remote_dte_address>
DTE Address Protocol Id = <dte_address_protocol_id>
Accept PDN charges = <accept_pdn_charges>

--ERROR-- Trunk name <trunk_name> is not defined.

--ERROR-- Trunk name <trunk_name> is not an X.25 trunk.

--ERROR-- Trunk name <trunk_name> may not appear more than once in the list of specified trunks.

--ERROR-- Remote DTE address <remote_dte_address> may not appear more than once in the list of specified remote DTE addresses.

--ERROR-- The remote DTE address <remote_dte_address> contains invalid characters.

--ERROR-- The remote DTE address <remote_dte_address> has an invalid bracketed expression.

--ERROR-- DTE address protocol id <dte_address_protocol_id> may not appear more than once in the list of specified DTE address protocol ids.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_x25_incall trunk_name=X25_TRUNK remote_dte_addresses='123?45[5-7]65*' ..
  dte_address_protocl_id=(3,8..12) accept_pdn_charges=TRUE

The X.25 incall definition(s) have been added.

add_x25_incall trunk_name=X25_TRUNK remote_dte_addresses='123?45[5-7]65*' ..
  dte_address_protocl_id=(6,22..32) accept_pdn_charges=FALSE

The X.25 incall definition(s) have been added.


add_x25_incall trunk_name=X25_TRUNK remote_dte_addresses='123?45[5-7]65*' ..
  dte_address_protocl_id=(3,55..66) accept_pdn_charges=TRUE

 --ERROR-- The X.25 incall definition for trunk X25_TRUNK is already defined as follows:

 Remote DTE Addresses     123?45[5-7]65*
 DTE Address Protocol Id  3 , 8..12
 Accept PDN charges       TRUE


add_x25_incall trunk_name=X25_TRUNK remote_dte_addresses='123545[5-11]65*' ..
  dte_address_protocl_id=(3,8..12) accept_pdn_charges=TRUE

 --ERROR-- The remote DTE address <remote_dte_address> has an invalid bracketed expression.
```

ADD_X25_SWITCH (ADDXS)

Purpose

Defines a general purpose X.25 switch definition. A switch is from the perspective of the CDCNET network. The X.25 Gateway uses this switch definition to route incoming X.25 calls to another X.25 Gateway within the CDCNET network and ultimately out to a PDN or another X.25 trunk. This command allows CDCNET to relay between PDNs or other X.25 trunks with a minimal amount of network administration. The switch is transparent: the calling DTE address, facilities, and user data for the outgoing side of the switch are the same as on the incoming side of the switch. However, if the parameter NEW_CALLED_DTE_ADDRESS is specified, the called DTE address in the packet is replaced with the value of NEW_CALLED_DTE_ADDRESS. If the parameter NEW_CALLED_DTE_ADDRESS is not specified, no substitution occurs, and the called DTE address in the packet remains the same.

The switch decision is based only on the called DTE address. The switch definitions have higher precedence than other X.25 definitions. In other words, an incoming X.25 call is initially checked to see if it should be switched. If switching is not required, it can be delivered based on a local definition. This means it is possible to switch some X.25 AsyncTip traffic (forward it to another PDN) while processing other X.25 AsyncTip traffic locally. X.25 calls that are switched are subject to validation by the ADD_X25_INCALL command but are not subject to validation by the ADD_X25_GW_INCALL command.

CONS traffic can be switched through CDCNET without the use of this command by using an ADD_NON_CDC_ROUTING_ENTRY command on the outbound side of the switch. However, only user data is transparently passed through the switch using this method.

If the LOCAL_DTE_ADDRESS parameter has been specified on the DEFINE_X25_INTERFACE command for the outbound side of the switch, there is no verification for switched calls. Switched calls are sent out on the first defined X.25 trunk on the outbound side of the switch.

Format

ADD_X25_SWITCH
GATEWAY_NAME = name
CALLED_DTE_ADDRESSES = list 1..31 of string 1..31 or OTHER
PEER_VE_OUTCALL_TITLE = name
NEW_CALLED_DTE_ADDRESS = string 1..15

Parameters

GATEWAY_NAME (GN)

Specifies the name of the X.25 A-to-A Gateway that receives the incoming X.25 calls to be switched. The gateway must be previously defined. This represents the inbound side of the X.25 switch.

CALLED_DTE_ADDRESSES (CDA)

Specifies one or more called DTE addresses. A DTE address can contain wild card attributes (to associate the switch definition with multiple DTE addresses) as follows:

[D1-D2] Within a bracketed group, a range of digits can be specified. D1 represents the low end of the range; D2 represents the high end. Any value within the range can be used to match a digit in the position specified in the address. For example, 1234[6-7]59 matches two DTE addresses: 1234659 and 1234759.

[...] Within a bracketed group, any list of single digits can be specified. For example, 1234[179]59 matches three DTE addresses: 1234159, 1234759, and 1234959.

* Any single * can be used to match a string of digits, including the null string. For example, 1234* matches 1234, 12341, 1234562345, and so forth.

? Any single ? can be used to match a digit in the position specified in the address. For example, 1234?59 matches 10 DTE addresses: 1234059, 1234159, 1234259 ... 1234959.

If multiple ADD_X25_SWITCH definitions match a given DTE address, the call definition associated with the most specific string is used. For example, 123?56 is more specific than 123* in matching calling DTE address 123456.

The keyword OTHER defines a switch definition for DTE addresses that do not match any other switch definitions for a given X.25 gateway.

PEER_VE_OUTCALL_TITLE (PVOT)

Specifies a VE_OUTCALL_TITLE registered by the peer X.25 Gateway in a DEFINE_X25_GATEWAY command. This represents the outbound side of the X.25 switch.

NEW_CALLED_DTE_ADDRESS (NCDA)

Specifies a DTE address that is substituted for the called DTE address of the incoming call packet when this packet is forwarded. No wild card character is allowed for this parameter. The default is the original called DTE address value.

Responses

The X.25 switch definition(s) have been added.

--ERROR-- The X.25 switch definition for X.25 gateway <gateway_name> is already defined as follows: (see the command response example for the format in which this information is displayed.)

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <gateway_name> is not defined.

--ERROR-- Called DTE address <called_dte_address> may not appear more than once in the list of specified called DTE addresses.

--ERROR-- The called DTE address <called_dte_address> contains invalid characters.

--ERROR-- The called DTE address <called_dte_address> has an invalid bracketed expression.

--ERROR-- The new called DTE address may include only digits 0 through 9.

--FATAL-- Not enough memory available for required table space.

Examples

```
add_x25_switch gateway_name=X25_GW_SWITCH ..
  called_dte_addresses=('123?45[5-7]65*', '23456789' ) ..
  peer_ve_outcall_title=OUTBOUND_SWITCH
  new_called_dte_address='23556789'

The X.25 switch definition(s) have been added.


add_x25_switch gateway_name=X25_GW_SWITCH ..
  called_dte_addresses=('987?45[5-7]65*', '23456789' ) ..
  peer_ve_outcall_title=OUTBOUND_SWITCH_2

--ERROR-- The X.25 switch definition for X.25 gateway X25_GW_SWITCH
          is already defined as follows:

Called DTE Addresses     23456789
Peer VE Outcall title    OUTBOUND_SWITCH
New Called DTE Address   23556789

add_x25_switch gateway_name=X25_GW_SWITCH ..
  called_dte_addresses='123545[5-11]65*' ..
  peer_ve_outcall_title=OUTBOUND_SWITCH_2

--ERROR-- The called DTE address 123545[5-11]65* has an
          invalid bracketed expression.
```

ADD_X25_TERMINAL_GW_OUTCALL (ADDXTGO) DI Configuration Procedure/NETOU Command

Purpose

Defines a X.25 terminal gateway outcall definition. Outcall is from the perspective of the CDCNET network; the call is going out of the CDCNET network. It allows the network administrator to configure X.25 outcall information and associate it with a logical name (outcall name) to help the end terminal user establish foreign connections. Also, the outcall information maybe used to restrict access to certain DTE addresses that are defined by the network administrator.

Format

ADD_X25_TERMINAL_GW_OUTCALL
OUTCALL_NAME = name
REMOTE_DTE_ADDRESS = string 1..15
GATEWAY_NAME = name
LOCAL_DTE_ADDRESS = string 1..15
FACILITIES = string 1..126
USER_DATA = string 1..248

Parameters

OUTCALL_NAME (ON)

Specifies a logical name associated with the REMOTE_DTE_ADDRESS. The logical name must be unique for each X.25 terminal gateway. Optionally, this logical name is available to the terminal user to help establish a foreign connection when the associated remote DTE address is a complete address.

REMOTE_DTE_ADDRESS (RDA)

Specifies the X.25 address of the destination X.25 system. It is specified as a string of digits 0 through 9 or the wildcard character *. If the DTE address contains only digits, the associated outcall name is available to the terminal user during connection establishment. The wildcard character * alone allows you to enter any remote DTE address. You are not restricted to those remote DTE address(es) added by the network administrator. For example, RDA=*, implies that the user is authorized to enter any remote DTE address. RDA='302*', implies that the user is authorized to enter remote DTE addresses that begin with 302. If the wildcard character is specified, it must be the last character in the entered string. Only one wildcard character may be entered.

GATEWAY_NAME (GN)

Specifies the name of the X.25 terminal gateway which provides access to the remote application. The X.25 terminal gateway must be previously defined. Defaults to the previously defined X.25 terminal gateway when entered in a configuration file. It is required when the command is entered by the operator.

LOCAL_DTE_ADDRESS (LDA)

Specifies the X.25 address of a local X.25 trunk. It forces selection of a specific trunk over which interactive X.25 connections to a remote DTE are made. If specified, it is included in the call request passed to the X.25 Packet Level. It must match a LOCAL_DTE_ADDRESS value on the DEFXI command. The call request is rejected by the X.25 Packet Level if no trunk with a matching LDA is found or if the matching trunk is DOWN. If not specified, and the gateway has been defined on multiple trunks, the X.25 Packet Level selects the X.25 trunk. The parameter is specified as a string of digits 0 through 9.

FACILITIES (F)

Specifies the facilities options as defined by the CCITT X.25 Recommendation. This parameter must be specified in order to use the fast select facility or for calls with reverse charges. This parameter must be entered as a string of an even number of hexadecimal digits.

USER_DATA (UD)

Specifies an even-numbered string of hexadecimal digits that are included in the X.25 Call Request. The call is rejected if the concatenated string exceeds the maximum call user data field size. The maximum field size is 124 octets with the fast select facility and 12 octets without the fast select facility (exclusive of the X.25 Call Request packet protocol field). The X.25 terminal gateway automatically inserts the 4-byte protocol field at the beginning of the user data.

Responses

X.25 terminal gateway outcall <outcall_name> added.

--ERROR-- X.25 terminal gateway <gateway_name> is not defined.

--ERROR-- Remote_dte_address can not include <string>. A remote_dte_address may include only digits 0 thru 9 or *.

--ERROR-- Remote_dte_address may only include one wildcard character (*).

--ERROR-- Remote_dte_address can not have digit(s) following wildcard character (*).

--ERROR-- Outcall_name <name> already exists for X.25 terminal gateway <gateway_name>.

--ERROR-- Local_dte_address can not include <string>. A local_dte_address may include only digits of 0 thru 9.

--ERROR-- Facilities can only have an even number of hexadecimal digits.

--ERROR-- User_data can only have an even number of hexadecimal digits.

--FATAL-- Not enough memory available for required table space.

Examples In this example, the FACILITIES parameter options are reverse charging not requested and fast select not selected (to determine the packet size and window size, see the CCITT X.25 Recommendation book).

```
add_x25_terminal_gw_outcall on=solver ..
rda='311061200260' ..
gn=telenet_terminal_gateway ..
f='0100420707430202'

X.25 terminal gateway outcall SOLVER added.

add_x25_terminal_gw_outcall on=transpac_access ..
rda='208*' ..
gn=transpac_terminal_gateway ..

X.25 terminal gateway outcall TRANSPAC_ACCESS added.
```

The ADDXTGO command processor orders the outcall blocks based on the length of the remote DTE address string and the presence of the wildcard characters. Complete addresses are linked in descending size of the DTE address.

```
Configuration file:

addxtgo,on=name1,rda='111111' gn=gw
addxtgo,on=name2,rda='2222222' gn=gw
addxtgo,on=name3,rda='33333333' gn=gw
addxtgo,on=name4,rda='444444' gn=gw
```

The above outcalls are linked and checked for matching criteria in the following order:

```
name3       33333333
name2       2222222
name4       444444
name1       111111
```

Any remote DTE address that contains a wildcard character is linked after complete DTE addresses, in descending size of the entered DTE address.

```
Configuration file:

addxtgo,on=name5,rda='3110*' gn=gw
addxtgo,on=name6,rda='3110612*' gn=gw
addxtgo,on=name7,rda='3110408*' gn=gw
addxtgo,on=name8,rda='1*' gn=gw
addxtgo,on=name9,rda='*' gn=gw
addxtgo,on=name10,rda='*' gn=gw,lda='311040800146'
```

Results in the following:

```
name3       33333333
name2       2222222
name4       444444
name1       111111
name7       3110408*
name6       3110612*
name5       3110*
name8       1*
name10      *
name9       *
```

The outcall name NAME9 is never used to satisfy a matching DTE address since NAME10 is the same string size but was entered after NAME9 and therefore linked ahead of NAME9.

The concept of predefined outcall information mirrors the definition of X.25 Gateway outcall title blocks, except that instead of registering the title in the directory, the outcall information is retained locally by the X.25 terminal gateway.

CANCEL Commands 4

CANCEL Commands 4

This chapter provides complete descriptions of all CDCNET Network Operations CANCEL commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

CANCEL_CHANNEL_TRUNK (CANCT) (NOS MDI Only) NETOU Command

Purpose Cancels the configuration of a channel trunk. The trunk is identified by its logical name.

Format

```
CANCEL_CHANNEL_TRUNK
   TRUNK_NAME = name
```

Parameters

TRUNK_NAME (TN)

The logical name of the trunk, assigned by the DEFINE_CHANNEL_TRUNK command that configured the trunk.

Responses

CHANNEL trunk <name> cancelled.

--WARNING-- Trunk <name> is not defined.

--ERROR-- Trunk <name> active, cannot be cancelled.

--ERROR-- Trunk <name> is not a CHANNEL trunk.

--ERROR-- VE interface <interface_name> cannot be cancelled by a cancel_channel_trunk command.

--ERROR-- Channel trunk <trunk name> cannot be cancelled until NP interface <interface name> is cancelled.

Examples

```
cancel_channel_trunk trunk_name = cyber101_alt

CHANNEL trunk CYBER_101_ALT cancelled.
```

CANCEL_DEVICE_OUTCALL_SERVICE (CANDOS) NETOU Command

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.

NOTE

In future releases, this command will no longer be supported. You should start using the CANCEL_OUTCALL_GATEWAY command as soon as possible.

Format CANCEL_DEVICE_OUTCALL_SERVICE

Parameters None.

Responses Device Outcall Service cancelled.

--WARNING-- Device Outcall Service not defined.

Examples

```
cancel_device_outcall_service

Device Outcall Service cancelled.
```

CANCEL_EGP_PEER (CANEP)
NETOU Command

Purpose Cancels the definition of a peer previously defined by the DEFINE_EGP_PEER command. Reference the peer by its IP Address. The specified peer must be stopped and in an idle state before it can be cancelled.

Format

```
CANCEL_EGP_PEER
   IP_ADDRESS = list 4 of 0..255
```

Parameters

IP_ADDRESS (IA)

The IP address of the EGP peer gateway which is to be cancelled. This peer must have been previously stopped and in an idle state before this command executes.

Responses

EGP peer <ip_address> is cancelled.

--ERROR-- EGP peer <ip_address> is not defined.

--ERROR-- EGP peer <ip_address> is not stopped.

--ERROR-- EGP peer <ip_address> is not yet idle.

Examples

```
cancel_egp_peer ip_address = (128,5,0,3)

EGP peer 128.5.0.3 is canceled.
```

CANCEL_ETHER_NET (CANEN)
NETOU Command

Purpose Cancels the configuration of an Ethernet network. The network is identified by its logical name.

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

```
cancel_ether_net network_name=engin_bldg_net

Ethernet network ENGIN_BLDG_NET cancelled for trunk ENGIN_BLDG_TRUNK.
```

CANCEL_ETHER_TRUNK (CANET) NETOU Command

Purpose Cancels the configuration of an Ethernet trunk. The trunk is identified by its logical name. The Ethernet trunk cannot be cancelled unless the network has been previously cancelled.

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 <name> cancelled.

--WARNING-- Trunk <name> not defined.

--ERROR-- Ether trunk <trunk name> cannot be cancelled until the Ether network <network_name> is cancelled.

--ERROR-- Trunk <name> is not an ETHERNET trunk.

Examples

```
cancel_ether_trunk trunk_name=engin_bldg_nsouth

Ethernet trunk ENGIN_BLDG_SOUTH cancelled.
```

CANCEL_FILE_SUPPORT (CANFS) (NOS MDI Only) NETOU Command

Purpose Cancels support for access of the specified file types from a 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

```
cancel_file_support file_type=dump trunk_name=c170_trunk1

File Support for specified file_types is cancelled for trunk c170_trunk1
```

CANCEL_HDLC_NET (CANHN) NETOU Command

Purpose Cancels the configuration of an HDLC network. The network is identified by its logical name.

Format

```
CANCEL_HDLC_NET
   NETWORK_NAME = name
```

Parameters

NETWORK_NAME (NN)

The logical name of the network assigned by the DEFINE_HDLC_NET command.

Responses

HDLC network <network_name> cancelled for trunk <trunk_name>.

HDLC network <network_name> cancelled for trunks (<trunk_name> <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

```
cancel_hdlc_net network_name = menlo_park_network

HDLC network MENLO_PARK_NETWORK cancelled for trunk MENLO_PARK_TRUNK.
```

CANCEL_HDLC_TRUNK (CANHT)
NETOU Command

Purpose Cancels the configuration of an HDLC trunk. The trunk is identified 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-- Trunk <name> active, can not be cancelled.

--ERROR-- HDLC Trunk <trunk_name> cannot be cancelled until the HDLC network <network_name> is cancelled.

Examples

```
cancel_hdlc_trunk trunk_name = menlo_park_trunk_1

HDLC trunk MENLO_PARK_TRUNK_1 cancelled.
```

CANCEL_IP_HOST (CANIH)
NETOU Command

Purpose Cancels the definition of an Internet Protocol (IP) host and its associated routing information. The host is identified by its IP 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 address <ip_address> is cancelled.

--WARNING-- IP host address <ip_address> is not defined.

--ERROR-- IP address <ip_address> is invalid.

Examples

```
cancel_ip_host ip_address=(128,2,53,7)

IP address 128.2.53.7 is cancelled.
```

CANCEL_IP_NET (CANIN)
NETOU Command

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 1..4 of 0..255

Parameters

IP_NETWORK (IN)

The IP address of the network or subnetwork, 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 cancelled.

IP subnet <ip_network> is cancelled.

--WARNING-- IP Network <ip_network> is not defined.

--WARNING-- IP subnet <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

```
cancel_ip_net ip_network=(128,2,0,0)

IP network 128.2.0.0 is canceled.
```

CANCEL_IP_X25_PDN_HOST (CANIXPH) NETOU Command

Purpose Cancels the definition of an IP X.25 PDN host and its associated routing information. The host is identified by its IP address.

Format

```
CANCEL_IP_X25_PDN_HOST
   IP_ADDRESS = list 4 of 0..255
```

Parameters

IP_ADDRESS (IA)

Specifies the IP address of the host, as previously defined by the DEFINE_IP_X25_PDN_HOST command that configured the host. 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).

Responses

IP address <ip_address> is cancelled.

--ERROR-- IP address <ip_address> is invalid.

--ERROR-- IP host address <ip_address> is not defined.

Examples

```
cancel_ip_x25_pdn_host ip_address = (128,2,53,7)

IP address 128.2.53.7 is cancelled.
```

CANCEL_LINE (CANL)
NETOU Command

Purpose Cancels the configuration of a communication line or a URI line. The line is identified 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, can not be cancelled.

--ERROR-- The device for line <line_name> is active and down, line cannot be cancelled.

Remarks When a line active error occurs, the line must first be stopped using the STOP_LINE command before the line can be cancelled. When a device is active and down, the port testing must first be stopped using the STOP_PORT_TEST command before the line can be cancelled.

Examples

```
cancel_line line_name=engin_line_1

Line ENGIN_LINE_1 cancelled.
```

CANCEL_NAME_RESOLVER (CANNR)
NETOU Command

Purpose Causes the Domain Name Resolver to stop accepting requests from users.

Format CANCEL_NAME_RESOLVER

Parameters None.

Responses Domain Name Resolver is cancelled.

--WARNING -- Domain Name Resolver is not defined.

Examples

```
cancel_name_resolver

Domain Name Resolver is cancelled.
```

CANCEL_NP_INTERFACE (CANNI) (NOS MDI Only) NETOU Command

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

```
cancel_np_interface in=cyber_109

NP interface CYBER_109 cancelled for trunk $MCI2.
```

CANCEL_NP_TERMINAL_GW (CANNTG) NETOU Command

Purpose Cancels the configuration of an NP terminal gateway. Cancels all titles associated with the gateway.

Format

```
CANCEL_NP_TERMINAL_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

The logical name of the interface assigned by a DEFINE_NP_ TERMINAL_ GW command.

Responses

NP terminal gateway <gateway_name> cancelled.

--WARNING-- NP terminal gateway <gateway_name> is not defined.

--ERROR-- NP terminal gateway <gateway_name> is active. It must be stopped before being cancelled.

Examples

```
cancel_np_terminal_gw gn = ivt_109

NP terminal gateway IVT_109 cancelled.
```

CANCEL_OPERATOR_SUPPORT (CANOS) (NOS MDI Only) NETOU Command

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

```
cancel_operator_support trunk_name=c170_trunk1

Operator Support is cancelled for c170_trunk
```

CANCEL_OUTCALL_GATEWAY (CANOG) NETOU Command

Purpose Cancels the definition of the Outcall Gateway from a DI. Since there can be only a single instance of the Outcall Gateway in a DI, the command does not require any parameters to identify the Outcall Gateway being cancelled.

Format CANCEL_OUTCALL_GATEWAY

Parameters None.

Responses Outcall Gateway is cancelled.

--WARNING-- Outcall Gateway is not defined.

--ERROR-- Outcall Gateway must be stopped before cancelling.

Examples

```
cancel_outcall_gateway

Outcall Gateway is cancelled.
```

CANCEL_PASSTHROUGH_SERVICE (CANPS) NETOU Command

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.

NOTE

In future releases, this command will no longer be supported. You should start using the CANCEL_OUTCALL_GATEWAY command as soon as possible.

Format CANCEL_PASSTHROUGH_SERVICE

Parameters None.

Responses Passthrough service cancelled.

--WARNING-- Passthrough Service not defined.

Examples

```
cancel_passthrough_service

Passthrough service cancelled.
```

CANCEL_RECORDER_LOG_GROUP (CANRLG) (NOS MDI Only) NETOU Command

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 no longer records the cancelled group (the recorder DI no longer logs 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

```
cancel_recorder_log_group lg=catenet tn=c170_trunk1

Recorder log groups are cancelled for trunk c170_trunk1.
```

CANCEL_REMOTE_LINE_MONITOR (CANRLM)
DI Configuration/NETOU Command

Purpose Cancels the selected Remote Line Monitor.

Format CANCEL_REMOTE_LINE_MONITOR

Parameters None.

Responses Remote Line Monitor cancelled, $LIM <lim>_PORT <port>.

Remote Line Monitor not defined for this DI.

Examples

```
cancel_remote_line_monitor

Remote Line Monitor cancelled, $LIM5_PORT0.
```

CANCEL_REMOTE_LOAD_SUPPORT (CANRLS) NETOU Command

Purpose Cancels support of the CDCNET system for loading and dumping 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

Parameters None.

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, see the DEFINE_REMOTE_LOAD_SUPPORT command in this manual.

Examples

```
cancel_remote_load_support

Remote Load Support is cancelled.
```

CANCEL_SERVER_TELNET_GW (CANSTG) NETOU Command

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 cancelled 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 cancelled.

--WARNING-- Server TELNET gateway <gateway_name> is not defined.

--ERROR-- Server TELNET gateway <gateway_name> is active. It must be stopped before being cancelled.

Examples

```
cancel_server_telnet_gw gateway_name=gw_to_cyber

Server TELNET gateway GW_TO_CYBER is cancelled.
```

CANCEL_SLIP_GW (CANSG)
NETOU Command

Purpose Cancels the definition of a SLIP gateway and the associated SLIP host entries. Use the STOP_SLIP_GW command to stop the gateway before cancelling it with this command.

Format

```
CANCEL_SLIP_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

The logical name of the SLIP gateway defined by a DEFINE_SLIP_GW command. This parameter is required.

Responses

SLIP gateway <gateway_name> is cancelled.

--ERROR-- SLIP gateway <gateway_name> is not defined.

--ERROR-- SLIP gateway <gateway_name> is active. It must be stopped before being cancelled.

Examples

```
cancel_slip_gw gateway_name=slip_gw

SLIP gateway SLIP_GW is cancelled.
```

CANCEL_SLIP_HOST (CANSH) NETOU Command

Purpose Deletes the specified SLIP entry.

Format

```
CANCEL_SLIP_HOST
   HOST_NAME = name
   GATEWAY_NAME = name
```

Parameters

HOST_NAME (HN)

The name of the SLIP host entry to delete. This parameter is required.

GATEWAY_NAME (GN)

The logical name of the SLIP gateway for which this entry was used. This parameter is required.

Responses

SLIP host <host_name> is deleted from gateway <gateway_name>.

--ERROR-- SLIP host <host_name> not found.

--ERROR-- SLIP gateway <gateway_name> is not defined.

Examples

```
cancel_slip_host gateway_name=slip_gw host_name=host_1

SLIP host HOST_1 is deleted from gateway SLIP_GW.
```

CANCEL_SNMP_AGENT (CANSA)
NETOU Command

Purpose Cancels the SNMP Agent previously defined by the DEFINE_SNMP_AGENT command. As part of the cancellation, the communities previously added by ADD_SNMP_COMMUNITY are deleted.

Format **CANCEL_SNMP_AGENT**

Parameters None.

Responses The SNMP Agent has been cancelled.

--WARNING-- The SNMP Agent is not defined.

Examples

```
cancel_snmp_agent

  The SNMP Agent has been cancelled.
```

This page was intentionally left blank.

CANCEL_SOURCE_ALARM_GROUP (CANSAG) NETOU Command

Purpose Cancels the alarm group(s) that the Dependent Log ME belongs to.

Format

CANCEL_SOURCE_ALARM_GROUP
ALARM_GROUP = list 1..15 of name

Parameters

ALARM_GROUP (AG)

Specifies the alarm group or list of alarm groups to be cancelled from the Dependent Log ME alarm environment. Default is CATENET.

Responses

Source alarm groups cancelled.

--WARNING-- Specified source alarm groups cancelled. Source alarm group <name> was not defined.

--WARNING-- No source alarm groups defined.

--ERROR-- Duplicate alarm group <name> specified.

Examples

```
cancel_source_alarm_group alarm_group = catenet

Source alarm groups cancelled.
```

CANCEL_SOURCE_ALARM_MESSAGE (CANSAM) NETOU Command

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. See the online CDCNET Diagnostic Messages manual for the complete list of alarm messages and their identifier numbers.

Responses Source alarm messages cancelled.

Examples

```
cancel_source_alarm_message mn=(3,42..45,87)

Source alarm messages cancelled.
```

CANCEL_SOURCE_LOG_GROUP (CANSLG) NETOU Command

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 = 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

```
cancel_source_log_group log_group=catenet

Source log group cancelled.
```

CANCEL_TCPIP_GW (CANTG)
NETOU Command

Purpose Cancels the definition of an application interface gateway for TCP/IP applications such as FTP and SMTP. The gateway must be stopped using the STOP_TCPIP_GW command before it is cancelled 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 cancelled.

--WARNING-- TCP/IP gateway <gateway_name> is not defined.

--ERROR-- TCP/IP gateway <gateway_name> is active. It must be stopped before being cancelled.

Examples

```
cancel_tcpip_gw gateway_name=ftp_gw

TCP/IP gateway FTP_GW is cancelled.
```

CANCEL_UDP_INTERFACE (CANUI) NETOU Command

Purpose Cancels the definition of the user datagram protocol (UDP) interface layer.

Format CANCEL_UDP_INTERFACE

Parameters None.

Responses UDP interface is cancelled.

--WARNING-- UDP interface is not defined.

--ERROR-- UDP interface is active. It must be stopped before being cancelled.

Examples

```
cancel_udp_interface

UDP interface is cancelled.
```

CANCEL_USER_TELNET_GW (CANUTG) NETOU Command

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 cancelled.

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

```
cancel_user_telnet_gw gateway_name=gw_to_vax

User TELNET gateway GW_TO_VAX is canceled.
```

CANCEL_USER_VALIDATION (CANUV)
NETOU Command

Purpose Cancels the validation of users connected to CDCNET through this DI.

Format CANCEL_USER_VALIDATION

Parameters None.

Responses User validation is cancelled.

--WARNING-- User validation is not defined.

Examples

```
cancel_user_validation

User validation is cancelled.
```

CANCEL_X25_ASYNCTIP (CANXA)
NETOU Command

Purpose Cancels the X.25 asynchronous TIP service supported by the specified X.25 trunk(s).

Format

CANCEL_X25_ASYNCTIP
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

X.25 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

```
cancel_x25_asynctip trunk_name = telenet_2

X.25 AsyncTip support cancelled for specified trunks.
```

CANCEL_X25_GW (CANXG)
NETOU Command

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

```
cancel_x25_gw gn=telenet_gw

X.25 gateway TELENET_GW is cancelled.
```

CANCEL_X25_INTERFACE (CANXI)
NETOU Command

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

```
cancel_x25_interface in=telenet_2

X.25 interface TELENET_2 cancelled for trunk TELENET2.
```

CANCEL_X25_NET (CANXN)
NETOU Command

Purpose Cancels the configuration of a X.25 network. The network is identified 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

```
cancel_x25_net network_name=tymnet_net_1

X.25 network TYMNET_NET_1 cancelled for trunk
TYMNET_TRUNK_3.
```

CANCEL_X25_TERMINAL_GW (CANXTG) NETOU Command

Purpose Cancels the configuration of an X.25 terminal gateway.

Format

```
CANCEL_X25_TERMINAL_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

Logical name of the gateway assigned by a DEFXTG command.

Responses

X.25 terminal gateway <gateway_name> is cancelled.

--WARNING-- X.25 terminal gateway <gateway_name> is not defined.

--ERROR-- X.25 terminal gateway <gateway_name> is active. It must be stopped before being cancelled.

Examples

```
cancel_x25_terminal_gw gn = datapac_terminal_gateway

X.25 terminal gateway DATAPAC_TERMINAL_GATEWAY is cancelled.
```

CANCEL_X25_TRUNK (CANXT) NETOU Command

Purpose Cancels the configuration of an X.25 trunk. The trunk is identified 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> is not defined.

--ERROR-- Trunk <name> is not an X.25 trunk.

--ERROR-- X.25 trunk <trunk_name> cannot be cancelled until the interface <interface_name> is cancelled.

-- ERROR-- Trunk <name> active, can not be cancelled.

Remarks The X.25 interface for the trunk must be stopped by a STOP_X25_INTERFACE command and cancelled by the CANCEL_X25_INTERFACE command, before the trunk can be cancelled.

Examples

```
cancel_x25_trunk trunk_name=tymnet_trunk_1

X.25 trunk TYMNET_TRUNK_1 cancelled.
```

CHANGE Commands 5

CHANGE Commands 5

This chapter provides complete descriptions of all CDCNET CHANGE commands used for network configuration and operations. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

CHANGE_CDCNET_ADDRESS_PREFIX (CHACAP) DI Configuration Procedure/NETOU Command

Purpose Changes the prefix and/or size of the CDCNET NSAP address.

An address prefix identifies a group of NSAP addresses. Part of each address in this group, called the address prefix, has the same value. As implied by its name, the CDCNET NSAP address prefix identifies the value of a common part of all CDCNET NSAP addresses.

When this command executes from a configuration file, the prefix and/or NSAP address size change occurs when system configuration completes. However, in this case, the system goes through one additional reset if the address prefix or size specified in the CHACAP command differs from its previously configured value.

Format

CHANGE_CDCNET_ADDRESS_PREFIX
ADDRESS_PREFIX = string of 4..22
ADDRESS_SIZE = 11..20

Parameters

ADDRESS_PREFIX (AP)

Prefix of CDCNET NSAP addresses. Specify this value as a string of hexadecimal digits, and include all leading or trailing zeros. The address prefix value must include an even number of hexadecimal digits.

If this parameter is omitted, the CHACAP command does not change the previously configured or initialized address prefix.

ADDRESS_SIZE (AS)

Size, in bytes, of the CDCNET NSAP addresses. Specify this value as a decimal number between 11 and 20.

If this parameter is omitted, the CHACAP command does not change the previously configured or initialized address size.

Responses

CDCNET NSAP address prefix and address size are changed.

CDCNET NSAP address prefix is changed.

CDCNET NSAP address size is changed.

Change of NSAP addressing accepted. No values are changed.

--WARNING-- CDCNET NSAP address prefix and address size are changed. CDCNET NSAP address prefix was already <address_prefix>. CDCNET NSAP address size was already <address_size>.

--WARNING-- CDCNET NSAP address prefix and address size are changed. CDCNET NSAP address size was already <address_size>.

--WARNING-- CDCNET NSAP address prefix and address size are changed. CDCNET NSAP address prefix was already <address_prefix>.

--WARNING-- CDCNET NSAP address prefix is changed. CDCNET NSAP address prefix was already <address_prefix>.

--WARNING-- CDCNET NSAP address size is changed. CDCNET NSAP address size was already <address_size>.

--ERROR-- The address prefix <address_prefix> contains non-hex characters.

--ERROR-- The address prefix <address_prefix> string length is not even.

--ERROR-- The address size is not large enough to accommodate the address prefix and the fixed portion of the CDCNET NSAP address. The address size must be at least 9 octets greater than the address prefix string length divided by two.

--ERROR-- Change_cdcnet_address_prefix can not be executed through the Network Operations Facility. Change_cdcnet_address_prefix may only be executed through a Device Interface configuration file.

Remarks This command is not required in a DI configuration file. Values in battery-backed RAM for the address prefix CDCNET NSAP address size are used if this command is not in a DI's configuration file. These initial system values are summarized in the Software Release Bulletin for the release that includes OSI.

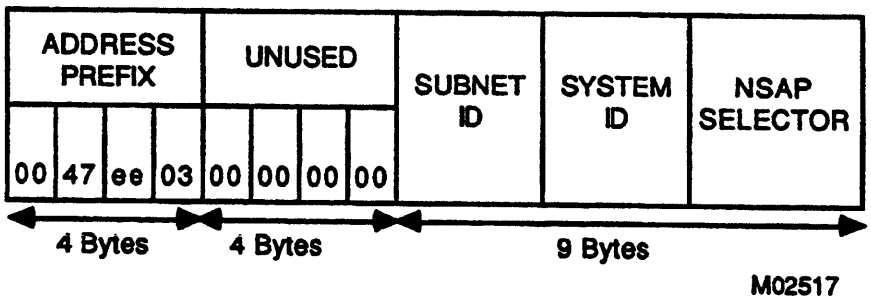
Examples

```
change_cdcnet_address_prefix ap='0447'

CDCNET NSAP address prefix is changed.

change_cdcnet_address_prefix as=17

CDCNET NSAP address size is changed.
```

The following examples further illustrate implications of changing the address prefix and/or CDCNET NSAP address size. Assume the following as initial or default values for the address prefix and CDCNET NSAP address size.

```
Size of the address prefix:   4 bytes
Value of the address prefix:  0047ee03(16)
Size of the NSAP address:     17 bytes
```

The following diagram represents these values for the CDCNET NSAP address.



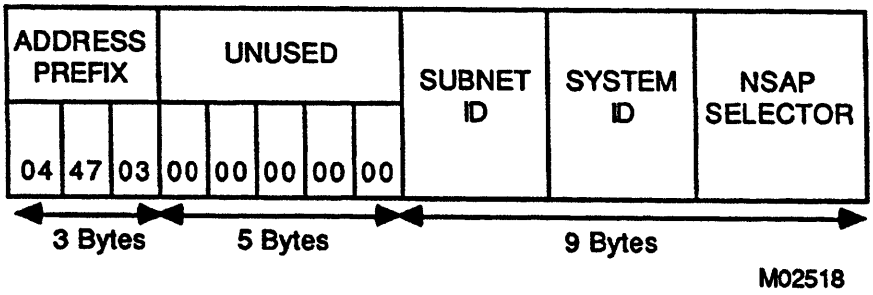
The command:

```
change_cdcnet_address_prefix ap='044703'
```

results in the following values for the address prefix, NSAP address size, and address format:

```
Size of the address prefix     3 bytes
Value of the address prefix    044703(16)
Size of the NSAP address       17 bytes
```

This command results in the following format for the CDCNET NSAP address:



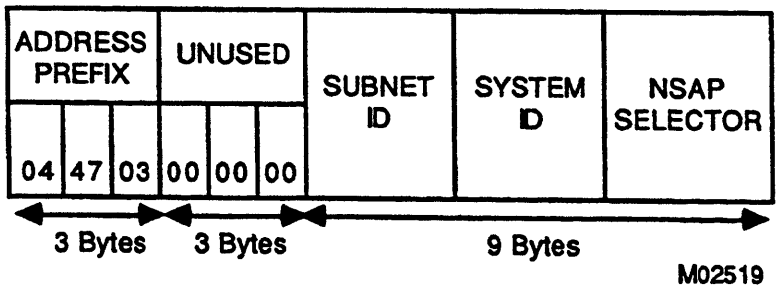
Assuming the above format to be the initial or current format of the CDCNET NSAP address, execution of the command:

```
change_cdcnet_address_prefix as=15
```

results in the following:

```
Size of the address prefix     3 bytes
Value of the address prefix    044703(16)
Size of the NSAP address       15 bytes
```

This results in the following format for the CDCNET NSAP address:



CHANGE_CHANNEL_TRUNK (CHACT) (NOS MDI Only) NETOU Command

Purpose Changes the channel timeout value of the channel trunk.

Format

```
CHANGE_CHANNEL_TRUNK
   TRUNK_NAME = name
   CHANNEL_TIMEOUT = boolean
```

Parameters

TRUNK_NAME (TN)

Logical name of the channel trunk.

CHANNEL_TIMEOUT (CT)

How MCI software handles host peripheral processor (PP) interruptions. This parameter is intended for use by customers during joint software development with Control Data. The following values are allowed:

Value	Description
TRUE	MCI software detects when a host PP stops operating, downs the channel interface, and attempts recovery.
FALSE	MCI software does not detect when a host PP stops operating. Specify FALSE only when debugging host PP software.

Default is not to change the current parameter value.

Responses CHANNEL trunk <trunk_name> options are changed.

--ERROR-- Trunk <trunk_name> is not defined.

--ERROR-- Trunk <trunk_name> is not a CHANNEL trunk.

Examples

```
change_channel_trunk trunk_name=C180_CHANNEL_1 channel_timeout
=FALSE

CHANNEL trunk C180_CHANNEL_1 options are changed.
```

CHANGE_CLNS (CHAC)
NETOU Command

Purpose Changes various operating characteristics of the connectionless mode network service (CLNS) entity.

Format

CHANGE_CLNS
HEADER_CHECKSUM = boolean
DERIVED_PDU_RETENTION_TIME = 1..500
PDU_MAXIMUM_LIFETIME = 1..256

Parameters

HEADER_CHECKSUM (HC)

Controls use the of checksum for the header of an outgoing CLNS initial PDU.

If TRUE, the CLNS entity calculates the checksum for the header that is prefixed onto the initial PDU. If the initial PDU needs to be segmented, the CLNS entity also checksums each derived PDU header.

If FALSE, the entity does not calculate the checksum for the header.

Initial value is FALSE.

DERIVED_PDU_RETENTION_TIME (DPRT)

Time, in seconds, the DI retains any derived PDUs it receives that do not form a complete initial PDU.

The CLNS entity combines the derived PDUs that it receives into the initial PDUs from which they were formed. If an initial PDU cannot be formed before the DPRT time expires, the derived PDUs for the initial PDU are discarded. The DPRT timer has a resolution of 5 seconds.

A small DPRT value ensures that DI resources is used only for a short time for PDUs that cannot be combined into a complete Initial PDU.

A large DPRT value ensures that PDUs transmitted over a slow medium are combined even when there is a long delay in receiving PDUs through that medium.

If all networks in the catenet are fast, like ETHERNET, then the DPRT value should be set low, for example, to 5 seconds. If the catenet includes a slow network, like an HDLC network running at 2400/4800 bps, then the DPRT value should be increased so that PDUs sent over the slow network are not discarded unnecessarily.

Initial value is 10 seconds.

PDU_MAXIMUM_LIFETIME (PML)

Maximum hops a network PDU can traverse. If a network PDU traverses more than the hops specified by this parameter, the PDU is discarded and an error PDU is sent back to the originator of the PDU. The error PDU states that the lifetime was exceeded. Set the value of PML to at least four times the number of hops in the longest path in the network. Initial value is 64 hops.

Responses CLNS environment updated.

Examples

```
change_clns header_checksum=TRUE

CLNS environment updated.
```

CHANGE_DIRECTORY (CHAD)
NETOU Command

Purpose Changes the version of the CDCNET Directory ME that the system uses.

NOTE

This command is optional in the DI configuration file.

Format

```
CHANGE_DIRECTORY
   VERSION = 2..3
```

Parameters

VERSION (V)

Specifies the Directory ME version used in transmitting directory PDUs. Both versions 2 and 3 are supported in receiving directory PDUs. Version 3 is more efficient than 2. Control Data recommends that you upgrade the entire catenet (including all NOS/VE hosts) before changing the version, since version 3 PDUs are not supported by older versions. The initial value for this parameter is 3.

Responses Directory M-E version changed.

CHANGE_ELEMENT_STATE (CHAES) NETOU Command

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)

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)

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> is not installed in system.

--ERROR-- Device <device_name> active. Stop communications or diagnostics before changing device state.

Examples

```
change_element_state device_name=$cim4,state=down

Device $CIM4 down.
```

CHANGE_ESIS_ROUTING_OPTIONS (CHAERO) DI Configuration Procedure/NETOU Command

Purpose Selects the OSI ES-IS Routing function to be supported in a CDCNET system. It allows you to change one or more configurable attributes of ES-IS routing support.

By default, each CDCNET system supports the end system and intermediate system routing functions. Therefore, you need this command only if the system needs to support the end system routing function. You can also use this command to change one or more attributes of the end or intermediate system routing function.

This command may be included in the configuration file or executed as an online operator command.

Format

CHANGE_ESIS_ROUTING_OPTIONS
ROUTING_FUNCTION = keyword
ES_CONFIGURATION_TIMER = 30..3600
IS_CONFIGURATION_TIMER = 10..1200
ROUTING_CACHE_INACTIVITY_TIMER = 60..1200
ENABLE_CHECKSUM = boolean
CONS_ROUTING_SUPPORT = boolean
ENABLE_CONFIGURATION_NOTIFY = boolean

Parameters *ROUTING_FUNCTION (RF)*

Changes the OSI routing functions present in the CDCNET system. One of the following keywords may be used:

Keyword	Description
ES	Selects the end system routing functions.
IS	Selects both end system and intermediate system routing functions.
COMBINED	Selects both end system and intermediate system routing functions.

In general, an ES has limited routing knowledge; IS and COMBINED have global routing knowledge. Communication between end systems on directly connected networks may be affected if a system resets an IS or COMBINED. The communication is fully restored either when the system that reset is reconfigured, or when its holding timer (derived from IS_CONFIGURATION_TIMER) expires in the end systems involved.

A system that has multiple network solutions should be COMBINED. A system that has at least one X.25 network solution should also be COMBINED because the ES-IS protocol is not designed to operate over PDNs.

If CONS is configured in a given system (by CHANGE_ESIS_ROUTING_OPTIONS or CHANGE_OSI_TRANSPORT), at least one system on each directly connected network must be configured as a COMBINED system. When an end system is defined that has CONS configured by the CHANGE_OSI_TRANSPORT command, all systems on the directly connected network that are configured as COMBINED systems must have CONS configured (by CHANGE_ESIS_ROUTING_OPTIONS or CHANGE_OSI_TRANSPORT).

The initial value of this parameter is COMBINED.

NOTE

The initial value is the value used for a given attribute when that attribute has not been specified in any previous use of this command.

ES_CONFIGURATION_TIMER (ECT)

Changes the value of the configuration timer used to control the rate at which an end system transmits End System Hello PDUs. A higher value reduces the CPU overhead but increases the time it takes other systems to learn about changes in the NSAP addresses present in the system or about this system having gone down. The value is specified in seconds. The initial value is 600.

IS_CONFIGURATION_TIMER (ICT)

Changes the value of the configuration timer used to control the rate at which an intermediate system transmits Intermediate System Hello PDUs. A higher value reduces the CPU overhead but increases the time it takes other systems to learn about a new intermediate system or loss of a currently known intermediate system. Control Data recommends that a small value be used for this timer. The value is specified in seconds. The initial value is 60.

ROUTING_CACHE_INACTIVITY_TIMER (RCIT)

Changes the value of the inactivity timer used to time out entries in the routing cache in an end system. A lower value for this timer increases the CPU overhead but deletes inactive entries sooner; therefore minimizing memory use. The value is specified in seconds. The initial value is 300.

ENABLE_CHECKSUM (EC)

Enables or disables generation of the checksum for ES-IS Routing PDUs transmitted by the local system. The initial value is FALSE.

CONS_ROUTING_SUPPORT (CRS)

Enables or disables the routing support for CONS. In intermediate or combined systems, it is used to enable or disable the software needed to support the relay function for CONS.

When the value of this option is changed from FALSE to TRUE, the software needed to support the relay function for CONS is loaded and started. This includes CONS IWU (CONS Interworking Unit), CONS, LLC 2 (Logical Link Control 2), and X.25 PL (X.25 Packet Level) software.

NOTE

It is possible that CONS, LLC 2, and X.25 PL software may already be loaded and started in order to support Transport over CONS (see the CHANGE_OSI_TRANSPORT command) or for X.25 PL, due to a DEFINE_X25_INTERFACE command.

When the value of this option is changed from TRUE to FALSE, CONS IWU quits using CONS to establish new connections. However, the existing connections over CONS are not affected. After all existing connections over CONS terminate, the software used to support the relay function for CONS stops and is made deloadable. However, CONS and LLC 2 are not stopped if they are needed to support Transport. X.25 PL is not stopped if it is needed to support Transport or if a DEFXI had previously been executed.

The initial value is FALSE.

ENABLE_CONFIGURATION_NOTIFY (ECN)

Enables or disables the configuration notification function of the ES-IS protocol. If enabled, it allows for rapid configuration of the routing database on a system that is initializing. The main disadvantage of enabling is that there may be cases when all the systems on an ESCI network flood the system being "helped" with ES-IS PDUs.

If this function is disabled, configuration may be delayed. The initial value of this parameter is TRUE.

Responses The following header precedes the routing options message.

ES-IS Routing options are changed.

Routing Function was already <routing_function>.
CONS Routing Support was already enabled.
CONS Routing Support was already disabled.

The following header is a success response.

ES-IS Routing options are changed.

The following messages are error responses.

--ERROR-- ES-IS Routing Options cannot be updated. CONS was unable to open an X.25 Packet Level SAP.

--FATAL-- Not enough memory is currently available for required table space.

Remarks CONS rejects all incoming calls requesting reverse charging.

If CONS is loaded and started, X.25 PL sends all incoming calls with no user data to CONS. If CONS is not loaded and started, X.25 PL sends all incoming calls with no user data to the X.25 PL user (potential users are X.25 Gateway, X.25 Network) that has defined a protocol ID of 0.

If a MAC-level bridge is used to connect two LLC 2 endpoints (for example, two CDCNET systems connected via an Ethernet bridge), then the bridge must ensure FIFO ordering, even if it utilizes multiple HDLC trunks. The LLC 2 protocol requires the underlying layers to preserve FIFO ordering.

CONS must be configured in every device between the source and destination endpoints of the associated OSI application. If there are multiple paths available, CONS must be configured in every device in each path. For example, if a NOS/VE host has two NOS/VE MDIs connected to a common Ethernet leading to an NDI with CONS configured, both NOS/VE MDIs must have CONS configured.

Examples

```
change_esis_routing_options ect = 60

ES-IS Routing options are changed.


change_esis_routing_options rf=ES ect = 60 crs=false

ES-IS Routing options are changed.
Routing function was already ES.
```

CHANGE_NAME_RESOLVER (CHANR)
NETOU Command

Purpose Changes any or all parameters specified in the DEFNR command. You do not need to cancel the Domain Name Resolver (DNR) first. Any parameter not reentered is not modified.

Format

CHANGE_NAME_RESOLVER
DOMAIN_NAME = string 1..255
SERVER_ADDRESSES = list 1..4 of 4 of integer 0..255
RETRY_LIMIT = integer 0..256
RETRY_TIMER = integer 1..32

Parameters

DOMAIN_NAME (DN)

Defines the name of the local administrative domain for the DNR. When requested to obtain information about a name and the name is a simple domain label, the DNR appends the DOMAIN_NAME parameter to the label before querying a Domain Name Server (DNS).

SERVER_ADDRESSES (SA)

Defines the order in which to query the IP addresses of the DNSs. The IP addresses are used in the order in which they are entered to communicate with the DNSs. At least one address is required which becomes the primary DNS. Up to three additional addresses may be entered specifying the secondary DNS.

RETRY_LIMIT (RL)

Defines the maximum number of times that a query is reissued over UDP.

RETRY_TIMER (RT)

Defines the timeout value used for UDP datagrams. It is used to reissue a UDP datagram if no response is received within the RETRY_TIMER period. Also, it is the value used to timeout a request if the application has not specified a timeout value.

Responses The TCP/IP Name Resolver is changed.

--ERROR-- Domain Name Resolver is not defined.

--ERROR-- Domain_name cannot contain the character <character>. Domain_name can only contain the characters A-Z, a-z, 0-9, hyphen (-) and underscore (_).

--ERROR-- A domain_name label is too long. Each label cannot be greater than 63 characters.

--ERROR-- UDP is not defined.

Examples

```
change_name_resolver domain_name = 'arh-cdc-com' ..
server_address = ((129,179,60,48) ..   (GRAY)
(129,179,52,203) ..   (ROYAL)
(129,179,48,30)) ..   (ODEON)
retry_limit = 5
retry_timer = 10

The TCP/IP Name Resolver is changed.
```

CHANGE_OSI_TRANSPORT (CHAOT)
NETOU Command

Purpose Changes various operating characteristics of the OSI Class 4 and where applicable, Class 0 Transport. These changes (from the first 4 parameters) affect connection establishment and apply only to connection establishment subsequent to command execution.

Format

CHANGE_OSI_TRANSPORT
CHECKSUMMING = boolean
ACKNOWLEDGE_TIMER = 0..30
MAXIMUM_WINDOW_SIZE = 1..16
MAXIMUM_CONNECTIONS = 0..65535 or *INFINITE*
ENABLE_OVER_CONS = boolean
CONS_USER_DATA = 2..32 or *NONE*

Parameters

CHECKSUMMING (C)

Determines how checksumming use is negotiated during connection establishment. This occurs for both incoming and outgoing connections, but does not apply to connections already established.

If the value is TRUE, the transport entity negotiates for checksumming; the peer must agree or disconnect.

If the value is FALSE, the transport entity attempts to negotiate not to use checksumming.

However, the peer may insist on checksumming. For example, if the peer resides in another DI that has this parameter set to TRUE, checksumming is performed. Checksumming prevents data corruption by certain types of errors, but increases processor load.

Initial value is FALSE.

ACKNOWLEDGE_TIMER (AT)

Represents the boundary for the maximum time that elapses between the receipt of a TPDU by the local transport entity, and the transmission of the corresponding acknowledge (ACK) TPDU. This value is sent to the peer during connection establishment.

A larger value for the acknowledge timer increases buffer use in systems that establish connections with the local system. This is because the sending transport must retain a copy of each message sent longer in case retransmission is necessary. However, the larger value lowers network and processor load by including acknowledges with other acknowledges, and reverse direction data messages. The effect is application-dependent with factors such as full-duplex versus half-duplex, or application "think" time.

Initial value is 5 seconds.

MAXIMUM_WINDOW_SIZE (MWS)

Specifies the upper boundary of user-specified transport window size. Transport users may specify maximum window size at connection establishment time. You can lower the initial value to save memory per connection. However, this increases network and processor load. A lower value decreases throughput on connections limited by the configured window size. A lower value can indirectly increase throughput on other connections by throttling connections limited by window size.

Initial value is 16.

MAXIMUM_CONNECTIONS (MC)

Maximum simultaneous OSI transport connections. If the value is INFINITE, there is no restriction on the number of connections other than the limitation imposed by physical memory available. It may be desirable to lower the initial value to reserve memory for executing status command processors and configuration changes. However, lowering this value also reduces the number of simultaneous terminal connections and file transfers.

Initial value is INFINITE.

ENABLE_OVER_CONS (EOC)

Controls the support of transport service over CONS. When TRUE is specified, service is supported.

When the value is changed from FALSE to TRUE, the software needed to support this service is loaded and started. This includes class 0 transport, CONS, Logical Link Control 2 (LLC2), and X.25 Packet Level (X.25 PL) software.

NOTE

CONS, LLC2, and X.25 PL software may have previously been loaded and started by the DEFXI command in order to support CONS interworking unit or the X.25 PL.

When the value of this parameter is changed from TRUE to FALSE, Transport software quits using CONS to establish new connections. However, existing connections remain unaffected. When all connections are terminated, software is stopped and is made deloadable.

NOTE

CONS, LLC2, and X.25 PL are not stopped if they are needed to support the CONS interworking unit, or the X.25 PL, due to a DEFXI command previously executed.

Initial value is FALSE.

CONS_USER_DATA (CUD)

Specifies the user data to be submitted to CONS on a connection request. The value must be a string containing an even number of hexadecimal digits or the keyword NONE.

A common use of CONS_USER_DATA among various vendors is to use the Network Management Connection Subprotocol (NMCS) UN PDU that signifies that the X.25 connection is dedicated for use by OSI Transport. This PDU is '03010100', four bytes long. The first byte, '03', is used as the protocol id. The DI version of CONS is set up to handle incoming X.25 connect requests destined for protocol id 3 or the null protocol id.

Initial value is NONE.

Responses The following headers are success responses.

OSI Transport is changed.

OSI Transport is changed. (Precedes one of the following lines)

Transport service over CONS was already enabled.
Transport service over CONS was already disabled.

The following are responses are error messages.

--ERROR-- Unable to change OSI Transport, enable over CONS = true due to CONS unable to open an X.25 Packet Level SAP.

--ERROR-- The CONS user data <cons_user_data> contains non-hexadecimal characters.

--ERROR-- The CONS user data <cons_user_data> contains an odd number of digits.

--FATAL-- Not enough memory is currently available for required table space.

Remarks CONS rejects all incoming calls requesting reverse charging.

If CONS is loaded and started, X.25 PL sends all incoming calls with no user data to CONS. If CONS is not loaded and started, X.25 PL sends all incoming calls with no user data to the X.25 PL user.

CONS must be configured in every device between the source and destination endpoints of the associated OSI application. If there are multiple paths available, CONS must be configured in every device in each path. For example, if a NOS/VE host has two NOS/VE MDIs connected to a common Ethernet leading to an NDI with CONS configured, both NOS/VE MDIs must have CONS configured.

Examples

```
change_osi_transport c=TRUE

OSI Transport is changed.
```

CHANGE_OUTCALL_GATEWAY (CHAOG) NETOU Command

Purpose Changes the outcall gateway default inactivity timer value. The timer value is specified in seconds.

Format **CHANGE_OUTCALL_GATEWAY**
INACTIVITY_TIMER = 120..14400 or *INFINITE*

Parameters *INACTIVITY_TIMER (IT)*

Adjusts the default inactivity timer. Users who do not transmit or receive data for a time period equal to this parameter's value are disconnected from the server and the server is made available to other users.

Responses Change of Outcall Gateway is accepted.

--ERROR-- Outcall Gateway is not defined.

Examples

```
change_outcall_gateway it=300

Change of Outcall Gateway is accepted.
```

CHANGE_PASSTHROUGH_SERVICE (CHAPS) NETOU Command

Purpose Changes the passthrough default inactivity timer.

NOTE

In future releases, this command will no longer be supported. You should start using the CHANGE_OUTCALL_GATEWAY command as soon as possible.

Format

CHANGE_PASSTHROUGH_SERVICE
INACTIVITY_TIMER = 120..14400 or *INFINITE*

Parameters *INACTIVITY_TIMER (IT)*

The inactivity timer measures the time during which no data is sent in either direction over a paired passthrough connection. This parameter specifies the maximum time, in seconds, a passthrough connection can be idle. When the time 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. The timer value is specified in units of seconds (range of 120..14400). Initial value is INFINITE.

Responses Change of Passthrough Service accepted.

--ERROR-- Passthrough Service not defined.

Examples

```
change_passthrough_service it=300

Change of Passthrough Service accepted.
```

CHANGE_PRINTER_MODEL_ATTRIBUTES (CHAPMA) DI Configuration Procedure/NETOU Command

Purpose Changes printer attributes previously set by a DEFINE_PRINTER_MODEL_ATTRIBUTES command for a specific printer terminal model. CHANGE_PRINTER_MODEL_ATTRIBUTES can be specified in a DI configuration file or via NETOU. There are no defaults for this command's parameters. If you do not specify a parameter, its value is left unchanged.

Format

CHANGE_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 = 2..12
VFU_TOP_FORM = boolean
MAXIMUM_VFU_LENGTH = 0..255
INITIALIZATION_PROCEDURE = name
INITIALIZATION_SEQUENCE = list 1..31 of <ccode>
EIGHT_LPI_SEQUENCE = list 1..7 of <ccode>
FILE_PREFIX_PROCEDURE = name
FILE_PREFIX_SEQUENCE = list 1..31 of <ccode>
FILE_SUFFIX_SEQUENCE = list 1..31 of <ccode>
SIX_LPI_SEQUENCE = list 1..7 of <ccode>

NOTE

For character code (ccode) values, see of the CDCNET Configuration Guide.

Parameters

TERMINAL_MODEL (TM)

The 1- through 31-character name of the printer terminal model for which attributes are being defined. This parameter may not be the same as any terminal model already defined by Control Data. It must have already been specified in a DEFINE_PRINTER_MODEL_ATTRIBUTES command.

AUTO_PAGE_EJECT_CHANNEL (APEC)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Channel causing the printer to skip automatically to the next top-of-form channel. May not be the same as BOTTOM_OF_FORM_CHANNEL (BOFC) value.

CHANNEL_1_SEQUENCE (C1S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "8" or "H" format effector is recognized in output lines.

CHANNEL_2_SEQUENCE (C2S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "7" or "G" format effector is recognized in output lines.

CHANNEL_3_SEQUENCE (C3S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "6" or "F" format effector is recognized in output lines.

CHANNEL_4_SEQUENCE (C4S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "5" or "E" format effector is recognized in output lines.

CHANNEL_5_SEQUENCE (C5S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "4" or "D" format effector is recognized in output lines.

CHANNEL_6_SEQUENCE (C6S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "3" or "C" format effector is recognized in output lines.

CHANNEL_7_SEQUENCE (C7S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "9" or "I" format effector is recognized in output lines.

CHANNEL_8_SEQUENCE (C8S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "X" or "J" format effector is recognized in output lines.

CHANNEL_9_SEQUENCE (C9S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "Y" or "K" format effector is recognized in output lines.

CHANNEL_10_SEQUENCE (C10S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "Z" or "L" format effector is recognized in output lines.

CHANNEL_11_SEQUENCE (C11S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "W" or "M" format effector is recognized in output lines.

CHANNEL_12_SEQUENCE (C12S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "U" or "N" format effector is recognized in output lines.

FORM_FEED_DELAY (FFD)

Supported by ASYNC and X.25 asynchronous TIPs only. Milliseconds (maximum 3000) the TIP must delay after sending a CHANNEL_x_SEQUENCE or FORM_FEED_SEQUENCE to the printer. The TIP sends null characters to the device to effect the delay.

FOLD_LINE (FL)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Indicates if the TIP must fold lines longer than the device page width.

FORM_FEED_SEQUENCE (FFS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "1" or "A" format effector is recognized in output lines.

KEYBOARD (K)

Supported by ASYNC and X.25 asynchronous TIPs only. Indicates if the printer has an associated keyboard.

If an asynchronous printer is configured with a keyboard, you can use the keyboard to enter the following commands to control the printer: DROP, START, and STOP. See the CDCNET Batch Device User's Guide for more information on these commands.

NO_SPACE_SEQUENCE (NSS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "+" format effector is recognized in output lines.

SINGLE_SPACE_DELAY (SSD)

Supported by ASYNC and X.25 asynchronous TIPs only. Milliseconds (maximum 1000) the TIP must delay after sending a SINGLE_SPACE_SEQUENCE to the printer. The TIP sends null characters to the device to effect the delay.

SINGLE_SPACE_SEQUENCE (SSS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when one of the following format effectors is recognized in output lines:

Format Effector	Times Single Space Sequence Sent
" " (blank; single space)	One
"0" (double space)	Two
"-" (triple space).	Three

BOTTOM_OF_FORM_CHANNEL (BOFC)

Channel to skip to when a "2" or "B" format effector is recognized in output lines. May not be the same as AUTO_PAGE_EJECT_CHANNEL (APEC) value.

VFU_TOP_FORM (VTF)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Defines whether of not the printer must be at top-of-form when the VFU is loaded.

MAXIMUM_VFU_LENGTH (MVL)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Maximum lines the printer supports in a VFU load image.

INITIALIZATION_PROCEDURE (IP)

Name of the load procedure containing data output to the printer when the line activates. Supported for any TIP handling transparent data.

INITIALIZATION_SEQUENCE (IS)

Sequence of octets sent to the printer when the printer activates and can receive output. If an initialization procedure is specified, the initialization sequence is sent after procedure data. Supported for any TIP handling transparent data.

EIGHT_LPI_SEQUENCE (ELS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "T" format effector is recognized in output lines. This parameter is allowed only if the DEFBD command for the printer has the parameter value VFU_LOAD_OPTION=NONE.

FILE_PREFIX_PROCEDURE (FPP)

Name of the load procedure containing data output to the printer before each nontransparent file. Supported for any TIP handling transparent data.

NOTE

If you change an FPP, do the following to guarantee that all printers and printer terminal models using the FPP get a copy of the changed FPP:

for each printer terminal model that uses the FPP, send a CHANGE_PRINTER_MODEL_ATTRIBUTES command to the DI or DIs supporting the printers. Specify the TERMINAL_MODEL and FILE_PREFIX_PROCEDURE parameters. For more information, see the CDCNET Configuration Guide.

FILE_PREFIX_SEQUENCE (FPS)

Sequence of octets sent to the printer before each file. If a file prefix procedure is specified, the file prefix sequence is sent after procedure data. Supported for any TIP handling transparent data.

FILE_SUFFIX_SEQUENCE (FSS)

Sequence of octets sent to the printer after each file. Supported for any TIP handling transparent data.

SIX_LPI_SEQUENCE (SLS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "S" format effector is recognized in output lines. This parameter is allowed only if the DEFBD command for the printer has the parameter value VFU_LOAD_OPTION=NONE.

Responses Printer model <name> changed.

--ERROR-- Cannot locate the specified printer terminal model.

--ERROR-- Printer model <name> is CDC defined-it cannot be modified.

--ERROR-- The bottom of form and auto page eject channels cannot be the same.

--ERROR-- If EIGHT_LPI_SEQUENCE or SIX_LPI_SEQUENCE is specified, then both must be specified.

Remarks If an EIGHT_LPI_SEQUENCE or a SIX_LPI_SEQUENCE is defined for a printer terminal model, both must be specified. These sequences may only be used for printers defined with the DEFINE_BATCH_DEVICE parameter value VFU_LOAD_OPTION=NONE.

Examples

```
change_printer_model_attributes tm=user_printer_model ..
  fold_line=true

Printer model user_printer_model changed.
```

CHANGE_SERVER_TELNET_GW (CHASTG) NETOU Command

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 = name
   TRANSLATION_DOMAIN = CATENET
   MAX_CONNECTIONS = 0..65535 or INFINITE
   TCP_PORT_NUMBER = 0..65535
   TCP_ALLOCATE_SIZE = 0..2147483647
   TCP_TIMEOUT = 0..65535 or INFINITE
   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. Server TELNET issues a TCP PASSIVE_CONNECT request using the well-known port for the source port. Initial value 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 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 never aborts.

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

```
change_server_telnet_gw gateway_name=gw_to_cyber ..
title=ivt_gateway max_connections=5

Server TELNET gateway GW_TO_CYBER is changed.
```

CHANGE_SERVICE_DISPLAY (CHASD)
DI Configuration Procedure/NETOU Command

Purpose Manages service list shown in the service availability display. Effects of multiple change commands is cumulative. The initial service list is empty; that is, if no CHANGE_SERVICE_DISPLAY command is entered, no services are displayed when users enter the DISPLAY_SERVICES command.

Format

CHANGE_SERVICE_DISPLAY
ADD_SERVICES = list 1..16 of name
DELETE_SERVICES = list 1..16 of name or *ALL*
STATUS_INTERVAL = 1..60 or *INFINITE*

Parameters

ADD_SERVICES or *ADD_SERVICE (AS)*

List of interactive service names (for example, NOS or NOS/VE, passthroughs) to add to the service availability display.

DELETE_SERVICES or *DELETE_SERVICE (DS)*

List of interactive service names to delete from the service availability display. If a service is specified on both ADD_SERVICES and DELETE_SERVICES, the DELETE_SERVICES parameter takes precedence. ALL specifies to delete all services from the list.

STATUS_INTERVAL (SI)

How often the status of services in the service availability display is updated.

The status of each displayable service is updated when a user enters the first DISPLAY_SERVICES command. A CREATE_CONNECTION command also updates status of a displayable service. If STATUS_INTERVAL has not expired, status of a displayable service is not updated when the next DISPLAY_SERVICES command is entered. The interval is in units of minutes. Initial value is 5 minutes.

INFINITE specifies that the interval never expires; the status of a service is updated only when a CREATE_CONNECTION command is entered.

CREATE_CONNECTION updates status only for the service name specified on CREATE_CONNECTION. That is, if a service has alternative names, their status is not automatically updated. As a result, conflicts in the status of an interactive service known by multiple names can occur 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.

Remarks Executing the DISS command can cause excessive traffic in larger network configurations. When traffic becomes to excessive, status of some of the displayable services may not be reported.

One method used to reduce network traffic is to configure no more than three displayable services in each TDI.

Another method used to reduce network traffic is to use the keyword value INFINITE on the STATUS_INTERVAL parameter of the CHASD command. This keyword value instructs the DISPLAY_SERVICES command to **not** use the CDCNET directory services to obtain the status of displayable services. (Using the directory services causes the congestion problem.)

When INFINITE is used, all services can be displayed safely. However, using INFINITE reduces accuracy of the reported status, because status of displayable services is only updated based on success or failure of users' CREATE_CONNECTION attempts for the services.

Examples

```
change_service_display add_service=veiaf

Services added to the displayable list.

          veiaf
```

CHANGE_SERVICE_DISPLAY_TEXT (CHASDT) DI Configuration Procedure/NETOU Command

Purpose Defines text shown in the service availability display. This text is displayed when a terminal user enters the DISPLAY_SERVICES command. If multiple services are specified, the same information applies to each service. You must specify at least one of the optional text parameters to change the text displayed for a service.

Format

CHANGE_SERVICE_DISPLAY_TEXT
SERVICE = list 1..16 of name or ALL
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 each value can be four 72-character strings. Since CDCNET commands are restricted to 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)

List of interactive service names for which the text applies. ALL specifies the text applies to all interactive services.

TEXT (T)

Text displayed if a service is up or busy. This text appears if the service is down and no DOWN_TEXT or TEMPORARY_DOWN_TEXT is defined. There can be up to four lines of text.

DOWN_TEXT (DT)

Text displayed when a service is down. There can be up to four lines of down text. It appears only if no TEMPORARY_DOWN_TEXT is defined.

TEMPORARY_DOWN_TEXT (TDT)

Text displayed when a service is down. There can be up to four lines of temporary down text. It is deleted when the service status changes from down to up. One use of this parameter is to send CHASDT through NETOU to enter messages for down services.

Responses Services information changed for services.

<service_name>
:
<service_name>

--ERROR-- At least one text parameter must be specified to cause the service display text to be changed.

--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.'

Services information changed for services.

          veiaf
```

CHANGE_SNMP_AGENT (CHASA)
NETOU Command

Purpose Changes operational parameters for the Simple Network Management Protocol (SNMP) agent. Only CDCNET allows commands of length less than 256. You can execute CHANGE_SNMP_AGENT as many times as needed to ensure all variables fit on the command line. SNMP Managers granted read-write access can also modify values.

Format

CHANGE_SNMP_AGENT
NAME = string 1..226
CONTACT = string 1..226
LOCATION = string 1..226
MAX_PDU_SIZE = integer 484..8192

Parameters

NAME (N)

Specifies the administratively assigned name for this communications device.

CONTACT (C)

Specifies the contact person for this communications device, together with information on how to contact the person.

LOCATION (L)

Specifies the physical location of this communication device.

MAX_PDU_SIZE (MPS)

Specifies the maximum size PDU that an SNMP agent builds and sends to the peer SNMP Manager.

Responses

The SNMP Agent has been changed.

--ERROR-- Source log group <name> is not defined.

--WARNING-- No message numbers specified.

--WARNING-- The SNMP Agent has not been changed. No parameters were specified.

--WARNING-- The SNMP Agent is not defined. Source log group changed.

Examples

```
change_snmp_agent location='Arden Hills, room 2N150'

  The SNMP Agent has been changed.

change_snmp_agent

  --WARNING-- The SNMP Agent has not been changed.  No parameters were specified.

change_snmp_agent location='locked shortlook configuration'

  --WARNING-- The SNMP Agent is not defined.
```

CHANGE_SOURCE_LOG_GROUP (CHASLG)
DI Configuration Procedure/NETOU Command

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 are defined and then all log message numbers specified by the DELETE_MESSAGE_NUMBER parameter are cancelled. Changes made by this command remain in effect until the next DI reload.

Format

CHANGE_SOURCE_LOG_GROUP
LOG_GROUP = 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. Initial value 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, see the online 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, see the online Diagnostic Messages manual.

Responses

Source log group changed.

--WARNING-- No message numbers specified.

--ERROR-- Source log group <name> is not defined.

Examples

```
change_source_log_group amn=(40,346,500)

Source log group changed.


change_source_log_group dmn=346

Source log group changed.
```

CHANGE_SYSTEM (CHAS)
DI Configuration Procedure/NETOU Command

Purpose

Changes the value of several parameters on the DEFINE_SYSTEM command. Changes memory and buffer allocation boundaries for a DI's memory management functions. Changes whether the system broadcasts IS-IS routing protocol data units or provides the catenet master clock through the Independent Clock ME. Changes the maximum number of recoveries before resetting the DI.

For DIs connected to a NOS host, this command also provides the master clock for the catenet through the Independent Clock ME. Finally, the command changes the maximum number of recoveries allowed before resetting the DI.

If CHANGE_SYSTEM is included in a DI configuration procedure, 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.

All changes except for the DATA_BUFFER_SIZE and RESERVED_SYSTEM_SPACE become effective immediately. Changes for these parameters become effective at the next system load. 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 1..3 of 1..99
MEMORY_BOUNDARY_PERCENTAGE = list 1..3 of 1..99
MEMORY_MANAGER_PERIOD = 1..10
RESERVED_SYSTEM_SPACE = 1000..32768
STANDARD_STACK_SIZE = 2048..8192
DEFAULT_CHANNEL_TRUNK = name
ROUTING_SYSTEM = boolean
CLOCKING_SYSTEM = boolean
MAXIMUM_RECOVERIES = 0..255
COMPRESS_DUMP = boolean

Parameters

DATA_BUFFER_SIZE (DBS)

Size, in bytes, of the system data buffers. The value of this parameter is stored in battery-backed non-volatile memory and the effects are not realized until a reset other than a power-on reset occurs.

The actual buffer size saved 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

Default is 144 for a DI with less than 4 megabytes of SMM and 524 for an ICA-II with 4 or more megabytes of SMM (set at power-up but can be altered if this parameter is specified).

BUFFER_BOUNDARY_PERCENTAGE (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_PERCENTAGE (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. 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. This parameter is illegal for an ICA-II system.

MAXIMUM_RECOVERIES (MR)

Specifies the maximum number of task recoveries allowed in a 60-minute period before the DI is reset.

COMPRESS_DUMP (CD)

Requests a compressed dump (omits free memory and buffers) of DI memory. Possible parameter values are YES and NO.

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-- Clocking_system can not be specified for an ICA II system.

--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

```
change_system mbp=(70,80,90)

Change of system accepted.
```

CHANGE_TCP_INTERFACE (CHATI) NETOU Command

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..512 or *INFINITE*
QUIET_TIME = 0..10000
RETRANSMIT_STRATEGY = keyword value
RETRANSMIT_TIME = 0..65535
SECURITY_CHECKING = keyword value
TIME_TO_LIVE = 0..255
SYN_ACK_TIME = 2..10000

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.

ACK_PERCENTAGE (AP)

Specifies the percentage of the receive window that must be full before an acknowledgment is issued.

MAX_BUFFERS (MB)

Specifies the maximum number of data bytes that TCP holds for a connection for both directions of travel.

MAX_SEGMENT_SIZE (MSS)

Specifies the maximum segment size in bytes to be negotiated for each new connection.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous TCP connections. If INFINITE is entered, no restriction is placed on the number of 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.

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.

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.

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.

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.

SYN_ACK_TIME (SAT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgment of the initial SYN segment. The connection is aborted at timeout. A SAT time too small can cause connection disconnects before the remote host has time to respond. A large SAT time can cause users to wait long periods of time for an unavailable host. Default is 60 seconds.

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

```
change_tcp_interface accept_strategy=in_window ..
ack_percentage=75 max_buffers=512 max_connections=INFINITE

TCP options changed.
```

CHANGE_TELNET_SERVER (CHATS) NETOU Command

Purpose

Changes the operational parameters of a particular Telnet server. The original values were previously specified by the ADD_TELNET_SERVER command.

NOTE

Changes to the TDP, TUP, PFO, ION, and VU parameters take effect with new connections to the gateway but do not affect existing connections. All other changes take effect for existing connections as well.

Format

CHANGE_TELNET_SERVER
SERVER_NAME = name
TERMINAL_DEFINITION_PROCEDURE = name or *NONE*
TERMINAL_USER_PROCEDURE = name or *NONE*
PROCEDURE_FILE_OPTION = IP_ADDRESS_CONCATENATION or *none*
CONNECTION_CONNECT_TIMEOUT = 20..1000 or *INFINITE*
CONNECTION_DISCONNECT_TIMEOUT = 0..1000 or *INFINITE*
USER_CONNECTION_LIMIT = 1..16
INITIAL_OPTION_NEGOTIATION = list 0..3 of keyword or *NONE*
VALIDATE_USERS = boolean
TCP_TIMEOUT = 0..65535 or *INFINITE*

Parameters

SERVER_NAME (SN)

Specifies the name of the Telnet server to be changed.

TERMINAL_DEFINITION_PROCEDURE (TDP)

Specifies the TDP automatically executed for all users when they first access this Telnet server. NONE specifies that no TDP is to be automatically executed.

TERMINAL_USER_PROCEDURE (TUP)

Specifies the TUP automatically executed for all users when they first access this Telnet server. NONE specifies that no TUP is to be automatically executed. This parameter can only be used when the TDP parameter has a value of NONE.

PROCEDURE_FILE_OPTION (PFO)

Specifies one of two options for qualifying the procedure name specified by either the TDP or TUP parameter. Specifying IP_ADDRESS_CONCATENATION causes the remote IP address identified within the incoming connection to be concatenated with the name specified by either the TDP or TUP in effect for this server. If neither a TDP nor TUP is in effect for this server, this parameter is ignored. NONE selects no concatenation.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines how much time the Telnet user has to create the first $input/$output connection. If no connection is established within that time, the Telnet connection is terminated. INFINITE indicates an infinite time. The timer has a precision of +/-2 seconds.

CONNECTION_DISCONNECT_TIMEOUT (CDT)

Defines how much time the Telnet 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 user's connection is terminated. INFINITE indicates an infinite time.

USER_CONNECTION_LIMIT (UCL)

Defines the maximum number of interactive $input/$output connections allowed at any one time by a Telnet user.

INITIAL_OPTION_NEGOTIATION (ION)

Specifies the set of Telnet options to be negotiated (transmitted) by this Telnet server during the initial startup of all Telnet connections. Any subset of the keywords; WILL_SGA, WILL_ECHO, and DO_SGA may be specified or the single keyword NONE may be specified. A list containing the keyword NONE is treated as if NONE were the only keyword specified. The keyword NONE specifies that no Telnet option negotiation is to be initiated by this Telnet server.

VALIDATE_USERS (VU)

Specifies if the Telnet users accessing this Telnet server are validated when user validation is enabled by the DEFINE_USER_VALIDATION command. If VU is TRUE, users are validated when validation is enabled.

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. Entering a small value (less than a few seconds) may cause frequent and unnecessary loss of service during periods of network congestion. Entering a large value may cause users to wait a long time after the host or network has failed. If you enter INFINITE, the connection will not time out. The default value is 300 seconds.

Responses TELNET server <server_name> is changed.

--WARNING-- TELNET server <server_name> was not found.

Examples

```
change_telnet_server server_name = arhnet..
user_connection_limit = 5 validate_users = TRUE

TELNET server ARHNET is changed.
```

CHANGE_TELNET_SERVICE_NAME (CHATSN)

Purpose Changes the name of the Telnet service used when accessing Telnet from this DI. This service is accessed when a NOS/VE host application has issued a VTP Create CDCNET Connect request to a Telnet service. This is done, for example, when a terminal user executes the CREATE_TELNET_CONNECTION (TELNET) command on a NOS/VE system.

Format

```
CHANGE_TELNET_SERVICE_NAME
   SERVICE_NAME = name
```

Parameters

SERVICE_NAME (SN)

Specifies the name of the Telnet service to be used when accessing Telnet from this DI for the situations described above. The name must be the title of a User Telnet Gateway configured in the network. The default prior to an explicit CHANGE_TELNET_SERVICE_NAME is TELNET.

Responses

Telnet service name is changed to <service_name>.

--ERROR-- Telnet service name <service_name> is already defined.

CHANGE_USER_TELNET_GW (CHAUTG) NETOU Command

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 = CATENET
MAX_CONNECTIONS = 0..65535 or *INFINITE*
SOURCE_IP_ADDRESS = list 4 of 0..255
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..2147483647
TCP_TIMEOUT = 0..65535 or *INFINITE*
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 by the 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 never aborts.

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

User TELNET gateway <gateway_name> is changed.

--ERROR-- User TELNET gateway <gateway_name> is not defined.

Examples

```
change_user_telnet_gw gateway_name=gw_to_vax ..
title=(telnet_vax, telnet_unix) max_connections=10

User TELNET gateway GW_TO_VAX is changed.
```

DEFINE Commands 6

DEFINE Commands 6

This chapter provides complete descriptions of all CDCNET DEFINE commands used for network configuration and operations activities. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

DEFINE_BOOT_DEFAULTS (DEFBD) Exception List Command

Purpose Sets the default software version and dump criteria for DIs without a specific exception list entry. This command must be the first command in the exception list and cannot be omitted. This command also specifies default values for DEFINE_EXCEPTION_SYSTEM command parameters having the same name.

Ths command is valid only in an exception list. Do not use this command in a DI configuration procedure.

Format

DEFINE_BOOT_DEFAULTS
DEFAULT_VERSION = 0..0FFFF(16)
DELETE_DUMP_ON_ERROR = list 1..10 of 0..0FF(16)

Parameters

DEFAULT_VERSION (DV)

Version number of the CDCNET object library and boot file to load into DIs. This value consists of one through four hexadecimal digits. Any version number that starts with an alphabetic character must have a leading zero when it is entered in the exception list. The SET_VERSION_LEVEL installation procedure requires the abbreviated form (DV) of this parameter (see Remarks).

DELETE_DUMP_ON_ERROR (DDOE)

List of error conditions (besides power-on reset) that do not cause a DI to dump its memory before loading. Specify the error conditions as one- or two-digit hexadecimal numbers. Default is to dump the DI before loading for all conditions but power-on reset (reset code 00(16)) and the KILL_SYSTEM command with the DUMP parameter set to NO (reset code 33(16)).

Remarks Two CDCNET software items are assigned a version number:

- CDCNET software release. This version number is embedded in the CDCNET software and cannot be changed by a network administrator.
- CDCNET object library and boot file. This version number is used for the DEFAULT_VERSION parameter value on this command. It specifies the boot and object library to load into a DI.

To allow the SET_VERSION_LEVEL installation procedure to alter the default version level (see NOS Installation handbook and CDCNET Installation section of the NOS/VE Software Release Bulletin), use the abbreviated form of the DEFAULT_VERSION parameter (DV=*vvvv*(16) where *vvvv* is the software version level). Also, the NOS permanent file name of the exception list must be ELIST. The SETVL procedure does not alter entries in the DEFINE_EXCEPTION_SYSTEM command.

On NOS, this command has no effect for host-connected DIs (MDIs and MTIs). For host-connected DIs, you must use the INITDCN file described in the NOS Installation handbook.

Examples This example establishes version 0403 as the default software version to load.

```
define_boot_defaults dv=0403(16)
```

DEFINE_CHANNEL_TRUNK (DEFCT)
DI Configuration Procedure/NETOU Command

Purpose Defines the channel level interface to a NOS host.

NOTE

Do not use this command for an ICA-II or from a DI to a NOS/VE host. Instead, use DEFINE_VE_INTERFACE, described later in this chapter.

Format

DEFINE_CHANNEL_TRUNK
SLOT = 0..7
TRUNK_NAME = name
UPLINE_MESSAGE_TIMEOUT = 2..64
CHANNEL_TIMEOUT = boolean

Parameters

SLOT (S)

Board slot number of the MCI board.

If a DI has only one MCI, this parameter is optional. If a DI has more than one MCI, this parameter is required.

TRUNK_NAME (TN)

Logical name of the channel trunk. The default name is constructed using the SLOT parameter, as in $MCI7.

UPLINE_MESSAGE_TIMEOUT (UMT)

This parameter is no longer in use. If a value is specified, it is ignored.

CHANNEL_TIMEOUT (CT)

How MCI software handles host peripheral processor (PP) interruptions. This parameter is intended for use by customers during joint software development with Control Data. The following values are allowed:

Value	Description
TRUE	MCI software detects when a host PP stops operating, downs the channel interface, and attempts recovery.
FALSE	MCI software does not detect when a host PP stops operating. Specify FALSE only when debugging host PP software.

Default is TRUE.

Responses

CHANNEL trunk <trunk_name> defined.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- Board slot xx 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.

--ERROR-- Specified CHANNEL board is not on.

--ERROR-- Specified CHANNEL board is already in use.

--ERROR-- TRUNK <name> is the boot source.
<Parameter> cannot be redefined.

--ERROR-- Define_channel_trunk can not be executed in an ICA II system.

--FATAL-- Not enough memory is currently available for required table space.

Remarks

If this command is performed for the channel trunk over which the DI was loaded, it redefines the trunk with the specified options. To redefine the channel trunk over which the DI was loaded, this command must be in the configuration file. During reconfiguration, you can provide a new TRUNK_NAME. However, we recommend not changing the CHANNEL_TIMEOUT parameter.

MCIs in an MDI should always be placed in the highest slot position(s) in the MDI/MTI to ensure MPB provides correct priority of service.

Examples

```
define_channel_trunk slot=7,trunk_name=channel_trunk_2

CHANNEL trunk channel_trunk_2 defined.
```

DEFINE_DEVICE_OUTCALL_SERVICE (DEFDOS) (NOS/VE Only)
DI Configuration Procedure/NETOU Command

Purpose Installs the Device Outcall service in a DI. Use this command in the configuration files of all DIs with devices that can receive connections from host applications.

NOTE

In future releases, this command will no longer be supported. You should start using the DEFINE_OUTCALL_GATEWAY_STATUS command as soon as possible.

Format

DEFINE_DEVICE_OUTCALL_SERVICE
TITLE = name

Parameters *TITLE (T)*

Title of the device outcall service. Devices connect to the device outcall service using CREATE_CONNECTION, with the SERVICE_NAME parameter equal to this parameter value. Default is DEVICE_OUTCALL.

Responses 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 Desktop/VE uses the CDCNET Device Outcall service. DEFINE_DEVICE_OUTCALL_SERVICE is only available for use with Desktop/VE. The default TITLE parameter value (DEVICE_OUTCALL) must be used.

Examples

```
define_device_outcall_service title=device_out

Device Outcall Service device_out defined and started.
```

DEFINE_EGP_PEER (DEFEP)
DI Configuration Procedure/NETOU Command

Purpose

Configures parameters for a peer Exterior Gateway Protocol (EGP) gateway, in which CDCNET exchanges IP network routing information using the EGP. Normally, EGP reports all known IP networks (residing in the network routing tables) to all its peers. In some cases, a site might want to isolate an IP network from DDN access. This is done using ADD_EGP_REPORTED_NETS.

NOTE

Defaults for the protocol intervals, timeouts, thresholds, and adjust parameters are recommended by the protocol definition (see RFC 904). Use the defaults unless you have communications problems with a particular EGP peer. Before adjusting parameters, analyze the communications problems, network topology, and the EGP.

Format

DEFINE_EGP_PEER
IP_ADDRESS = list 4 of 0..255
AUTONOMOUS_SYSTEM_NUMBER = 1..65535
PEER_RECOVERY = keyword value
MINIMUM_HELLO_INTERVAL = 1..9999
MINIMUM_POLL_INTERVAL = 1..9999
REQUEST_CEASE_INTERVAL= 1..9999
HOLD_UP_DOWN_TIMEOUT = 1..9999
HOLD_ACQUISITION_CEASE_TIMEOUT = 1..9999
ACTIVE_NEIGHBOR_UP_THRESHOLD = 1..9
ACTIVE_NEIGHBOR_DOWN_THRESHOLD = 1..9
PASSIVE_NEIGHBOR_UP_THRESHOLD = 1..9
PASSIVE_NEIGHBOR_DOWN_THRESHOLD = 0..9
TRANSMIT_HELLO_ADJUST = 1..9999
TRANSMIT_POLL_ADJUST = 1..9999
RECEIVED_HELLO_ADJUST = 1..9999
RECEIVED_POLL_ADJUST = 1..9999
TEST_IP_NETWORK = boolean
START = boolean

Parameters

IP_ADDRESS (IA)

IP address of the EGP peer with which the neighbor acquisition protocol is exchanged. This IP host address must be on a directly connected wide area network (WAN) defined by DEFINE_IP_NET. The format is similar to the decimal octet convention used by TCP/IP, 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).

AUTONOMOUS_SYSTEM_NUMBER (ASN)

Autonomous system number of this local network, identifying IP networks this EGP gateway supports. This number is assigned by Stanford Research Institute (SRI).

PEER_RECOVERY (PR)

Peer recovery. Values include AUTO, MANUAL, and BACKUP. For a single peer, specify AUTO for backup support or MANUAL for no backup support.

Keyword Value	Description
AUTO (A)	When the start command (START_EGP_PEER) is received for the peer, EGP attempts to acquire the peer. If EGP cannot acquire the peer or loses the peer later, it attempts a BACKUP peer acquisition.
MANUAL (M)	When the start command (START_EGP_PEER) is received for the peer, EGP attempts to acquire the peer. If EGP cannot acquire the peer or loses the peer, it does not attempt a BACKUP peer acquisition.
BACKUP (B)	When the start command (START_EGP_PEER) is received for the peer, the peer is eligible to be acquired if the AUTO peer cannot be acquired or acquisition is lost. When the total active AUTO or MANUAL peers for a network falls below the total started AUTO peers, a BACKUP peer for that network initiates to fill the vacancy.

Default is AUTO.

MINIMUM_HELLO_INTERVAL (MHI)

Minimum interval, in seconds, acceptable between successive HELLO commands. This parameter is the EGP variable P1, the value reported to the EGP peer as the desired minimum interval between HELLO commands received from that peer. Default is 30 seconds.

MINIMUM_POLL_INTERVAL (MPI)

Minimum interval, in seconds, acceptable between successive POLL commands. This parameter is the EGP variable P2, the value reported to the EGP peer as the desired minimum interval between POLL commands received from that peer. Default is 120 seconds.

REQUEST_CEASE_INTERVAL (RCI)

Interval, in seconds, maintained between REQUESTS or CEASE command retransmissions. This is the EGP variable P3. Default is 30 seconds.

HOLD_UP_DOWN_TIMEOUT (HUDT)

Interval, in seconds, during which state variables are maintained in the absence of responses in the up and down states. This is the EGP variable P4. The interval is normally set to hold information while the peer dumps and restarts. Default is 3600 seconds.

HOLD_ACQUISITION_CEASE_TIMEOUT (HACT)

Interval, in seconds, during which state variables are maintained in the absence of responses in the acquisition and cease states. This is the EGP variable P5. Default is 120 seconds.

ACTIVE_NEIGHBOR_UP_THRESHOLD (ANUT)

Neighbor-up threshold used in the active mode of the neighbor reachability algorithm. Default is 3.

ACTIVE_NEIGHBOR_DOWN_THRESHOLD (ANDT)

Neighbor-down threshold used in the active mode of the neighbor reachability algorithm. Default is 1.

PASSIVE_NEIGHBOR_UP_THRESHOLD (PNUT)

Neighbor-up threshold used in the passive mode of the neighbor reachability algorithm. Default is 1.

PASSIVE_NEIGHBOR_DOWN_THRESHOLD (PNDT)

Neighbor-down threshold used in the passive mode of the neighbor reachability algorithm. Default is 0.

TRANSMIT_HELLO_ADJUST (THA)

Increases the interval, in seconds, between successive hello messages transmitted. Lengthening the hello interval can prevent violating the minimum hello interval because of bunching in the net. Default is 10.

TRANSMIT_POLL_ADJUST (TPA)

Increases the interval, in seconds, between successive poll messages transmitted. Lengthening the poll interval can prevent violating the minimum poll interval because of bunching in the net. Default is 20.

RECEIVED_HELLO_ADJUST (RHA)

Specified in seconds, this value is subtracted from the incoming hello interval in request or confirm messages. The result is the minimum acceptable interval between received hello messages. Default is 5 seconds.

RECEIVED_POLL_ADJUST (RPA)

Specified in seconds, this value is subtracted from the incoming poll interval in request or confirm messages. The result is the minimum acceptable interval between received poll messages. Default is 15 seconds.

TEST_IP_NETWORK (TIN)

Used only for testing. Normally, EGP peers must connect to the same IP network. This parameter configures a peer on a remote network. That is, it allows the peer to reside on an IP network not directly connected to the local EGP. Values include TRUE and FALSE. If the parameter is TRUE, the IP_ADDRESS parameter can be any IP network address and is not checked for an address directly connected to the local EGP. Default is FALSE.

START (S)

Whether or not the EGP gateway starts after configuration. Values are TRUE and FALSE. If START is TRUE and PEER_RECOVERY is either AUTO or MANUAL, EGP attempts peer acquisition immediately. If START is TRUE and PEER_RECOVERY is BACKUP, peer acquisition may be started if one of the other peer acquisitions fails. If START is FALSE, no action is taken for this peer definition until START_EGP_PEER is received. Default is TRUE.

Responses

EGP Gateway Peer <ip_address> is defined and started.

EGP Gateway Peer <ip_address> is defined.

--ERROR-- EGP Gateway Peer <ip_address> is already defined.

--ERROR-- EGP Gateway Peer <ip_address> is not on a directly connected IP Network.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_egp_peer ip_address=(10,61,0,2) ..
autonomous_system_number=78

EGP Gateway Peer 10,61,0,2 is defined and started.
```

DEFINE_ETHER_NET (DEFEN)
DI Configuration Procedure/NETOU Command

Purpose Configures a CDCNET Ethernet network solution using a previously defined Ethernet trunk. An "unable to start" error leaves the network defined but not started.

Format

DEFINE_ETHER_NET
TRUNK_NAME = name
NETWORK_ID = 1..0FFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
MULTICAST_NETWORK = boolean
ROUTING_INFO_NETWORK = boolean
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..500000
INCOMING_SERVICE_ACCESS = list 1..11 of keyword value
OUTGOING_SERVICE_ACCESS = list 1..11 of keyword value

Parameters

TRUNK_NAME (TN)

Logical name of the Ethernet trunk used for the network solution. The Ethernet trunk with this name must be configured by DEFINE_ETHER_TRUNK before this command executes.

NETWORK_ID (NI)

CDCNET network identifier of the Ethernet network solution. This number must be unique within the catenet.

NETWORK_NAME (NN)

Logical name of the network solution; used in subsequent commands referring to the network solution. The default name is constructed from the NETWORK_ID parameter, using the format $NET_*xxxxx*, where *xxxxx* is the network ID expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

Relative cost of the network solution as a path for routing data through the network. The default cost of an Ethernet trunk is 0A(16), the cost of a 10 M byte trunk. If the cost specified is greater than 0FFFF(16), 0FFFF(16) is used for the COST parameter value for compatibility reasons.

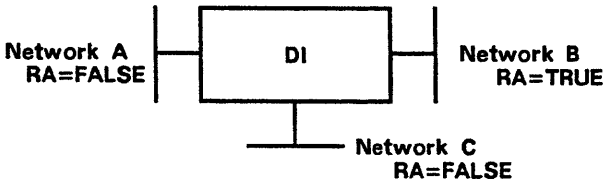
CDCNET routing uses cost to determine the least-cost routes for interconnecting networks. If multiple paths are available, the least-cost path is used. As a network becomes congested, its cost increases. It is extremely rare for an Ethernet network solution to become congested.

RELAY_ALLOWED (RA)

Whether or not relay is allowed through this network solution. Values are TRUE, relay allowed; and FALSE, relay not allowed. Default is TRUE.

If RA is TRUE, the DI relays traffic between this network and other connected networks regardless of the definition of the other networks. If RA is FALSE, the DI still relays traffic between this network and any connected networks defined with RA=TRUE. If RA is FALSE, the DI does not relay traffic between this network and other connected networks defined with RA=FALSE *unless* no other nonrestricted route exists between the networks.

The following example illustrates these rules.



Relay Between:	Allowed/Not Allowed:
Network A and Network B	Allowed (RA is TRUE for Network B).
Network A and Network C	Not allowed (RA is FALSE for both Network A and Network C).
Network B and Network C	Allowed (RA is TRUE for Network B).

MULTICAST_NETWORK (MN)

Whether or not the network solution is a multicast network. A multicast network supports sending a message to more than one device at a time. Values are TRUE and FALSE. Default is TRUE.

ROUTING_INFO_NETWORK (RIN)

Whether or not the network solution carries CDCNET routing information. Routing information pertains both to the proprietary CDCNET IS-IS routing protocol, as well as the ES-IS routing protocol. If RIN is TRUE, routing information describing all networks to which this system connects is sent over the network solution. If RIN is FALSE, routing information is not sent by this system over the network solution. This system appears unconnected to any network other than this network solution. Default is TRUE.

START (S)

Whether or not the network solution starts after configuration. Values are TRUE, start; and FALSE, do not start. Default is TRUE.

ARCHITECTURE_TYPE (AT)

This parameter is no longer supported. Specification is valid but the value is ignored.

OUTPUT_QUEUE_LIMIT (OQL)

Maximum data, in bytes, retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. If messages must be discarded, newer output messages are discarded first.

Default depends on network cost (see COST parameter). If the cost is 6FA(16) or more, default is 30000 bytes. Otherwise, default is 60000 bytes.

INCOMING_SERVICE_ACCESS (ISA)

Specifies which services may be accessed coming in via the network solution. The keyword values are:

Keyword Value	Description
ALL	All services are allowed.
NONE	No services are allowed.
application	Any service not identifiable by keyword batch_support, interactive, file_transfer, or network_management.
batch_support	Any batch input and output services are allowed. This includes batch control facility, batch job submittal, and batch output.
interactive	All interactive services are allowed. Interactive service supports terminal user access. Specifying this keyword is the same as specifying all of the following interactive service keywords: nos_interactive, passthrough, t_passthrough, user_telnet, x25_terminal_gw, and ve_interactive.
nos_interactive	All interactive services provided by NOS mainframes are allowed.
passthrough	All site-defined passthrough services are allowed. A site-defined passthrough service is provided when a passthrough title is created during the execution of a Terminal User Procedure (TUP) when a line is activated. This service allows asynchronous interactive devices to make connections to passthrough titles of this type through asynchronous ports and exchange information using CDCNET's transparent data feature.

Keyword Value	Description
t_passthrough	All user-defined passthrough services are allowed. A user-defined passthrough service is provided when an interactive terminal user creates a connection to the interactive passthrough server, and defines a passthrough title either by entering the required commands manually or by executing a Terminal User Procedure (TUP) via the DO command on the $CDCNET_COMMAND connection. This service allows asynchronous interactive devices to make connections to passthrough services of this type through asynchronous ports and exchange information using CDCNET's transparent data feature.
user_telnet	All TELNET services are allowed. User TELNET is an interactive service which provides CDCNET terminal users with access to the interactive services of remote hosts on a TCP/IP network.
x25_terminal_gw	All X.25 terminal services are allowed. The X.25 terminal gateway is an interactive service which provides CDCNET asynchronous terminal users with access to the interactive services of remote hosts on an X.25 network.
ve_interactive	All interactive services provided by NOS/VE mainframes are allowed.
file_transfer	All QTF/PTF file transfer services are allowed. File transfer services support user file (PTF) and queue file transfers (QTF).
network_management	All network alarm, clock, command, file, and log services are allowed.

Default is ALL.

OUTGOING_SERVICE_ACCESS (OSA)

Specifies which services may be accessed going out via the network solution. See the ISA parameter for the keyword values.

Responses

Ethernet network <network_name> defined for trunk <trunk_name>.

Ethernet network <network_name> defined and 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 <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> is down. Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> is off. Unable to start network <network_name>.

--ERROR-- NETWORK <name> is the boot source. <Parameter> cannot be redefined.

--ERROR--Network <network_name> already defined for
trunk <trunk_name> by the boot source.
Network id is <network_id>.
The command network id is <network_id>.
Please reenter the command with correct network id.

--FATAL-- Not enough memory is currently available 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.

Remarks This command is required only in DIs configured with an Ethernet network solution. It is not required in DIs booted over the Ethernet trunk. If this command is for the Ethernet trunk over which the DI boot was received, it redefines the network with the specified options. To redefine the Ethernet network over which the DI is booted, this command must be in the configuration file. Currently, redefinable options are:

NETWORK_NAME
COST
RELAY_ALLOWED
MULTICAST_NETWORK
ROUTING_INFO_NETWORK
OUTPUT_QUEUE_LIMIT
INCOMING_SERVICE_ ACCESS
OUTGOING_SERVICE_ACCESS.

Examples In this example, an NDI links two Ethernet network solutions (figure 6-1). Assume that service access between the two networks must be limited to the following:

- QTF/PTF file transfers in either direction.
- Interactive service in either direction.
- Host_1 provides network management and batch support for both networks. However, Host_2 provides those services only for ESCI_ Net_2.

Add the following to the configuration file for NDI_1:

```
define_ether_net..
   trunk_name=esci_trunk_1,network_name=esci_net_1..
   network_id=1..
   outgoing_service_access=(batch_support,file_transfer,interactive..
     network_management)..
   incoming_service_access=(file_transfer,interactive)

define_ether_net..
   trunk_name=esci_trunk_2,network_name=esci_net_2..
   network_id=2
```

Outgoing and incoming service access parameters are not included for esci_net_2 so service access defaults to ALL.

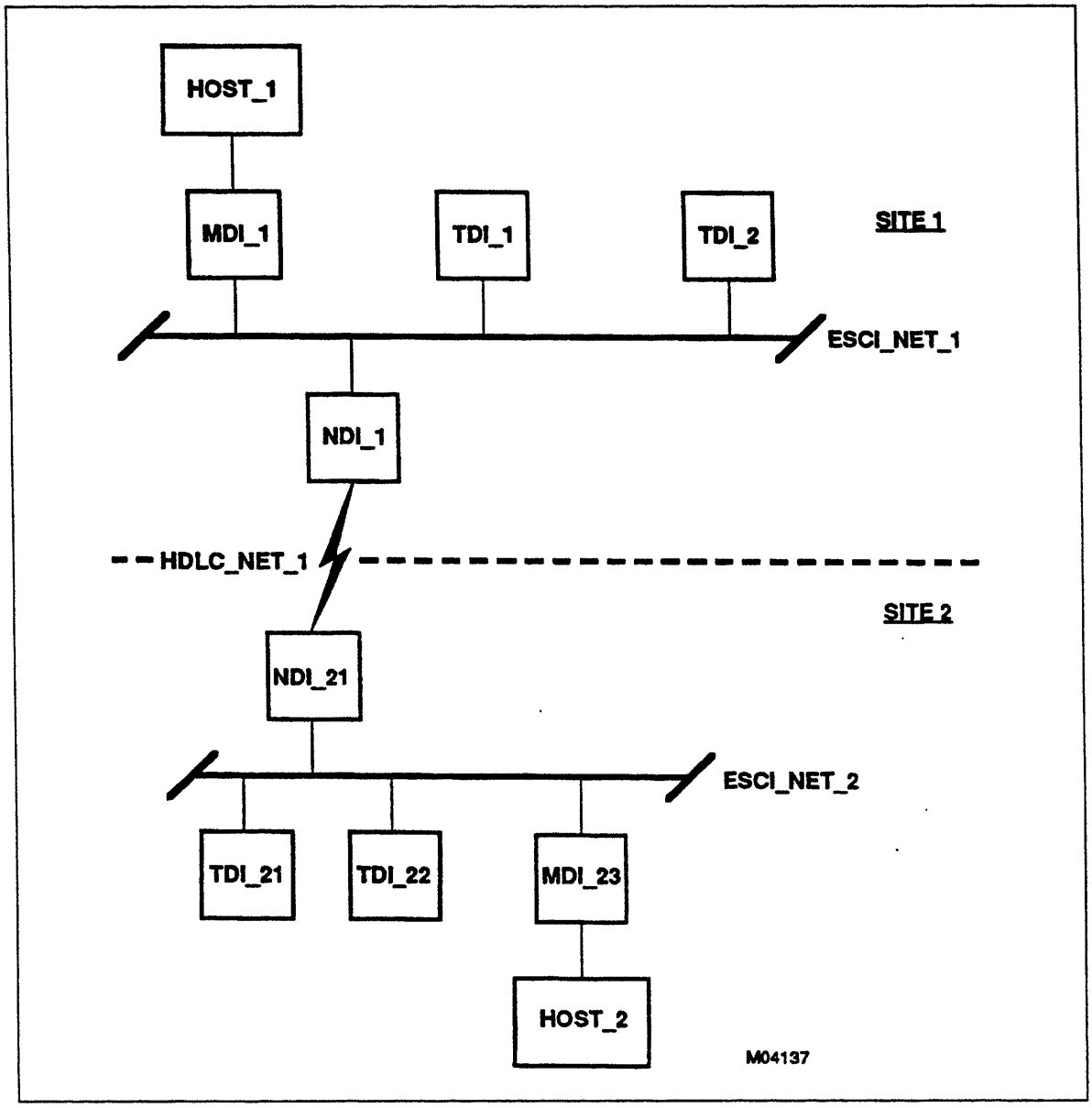


Figure 6-1. DEFEN Example

DEFINE_ETHER_TRUNK (DEFET)
DI Configuration Procedure/NETOU Command

Purpose Prepares an Ethernet cable to serve as a CDCNET trunk. Ethernet trunks used to load DIs are predefined by DI software. Executing this command for an already-defined trunk results in an error message. If a DI contains only one Ethernet trunk, you can omit this command from the DI's system configuration procedure, since the DI determines information required to define the trunk by the DI load process.

Format

DEFINE_ETHER_TRUNK
SLOT = 0..7
TRUNK_NAME = name
MAX_FRAME_SIZE = 132..1514
INTERFRAME_SPACING = 0..255

Parameters *SLOT (S)*

DI board slot number containing the Ethernet Serial Channel Interface (ESCI) board. If a DI has only one ESCI board, this parameter is optional. If a DI has more than one ESCI board, this parameter is required.

NOTE

Do not use this parameter for an ICA-II.

TRUNK_NAME (TN)

Logical name of the trunk. The default name is derived from the SLOT parameter, as in $ESCI3. For an ICA-II board, the default trunk name is $ICA_IEI. The trunk name must be unique in the DI.

MAX_FRAME_SIZE (MFS)

Maximum frame size, in bytes, the Ethernet can transmit over the Ethernet trunk. The value of this parameter must be coordinated with the DEFIN command if the DEFIP command configures an IP network over the trunk being configured. Default is 1514 bytes.

INTERFRAME_SPACING (IS)

Minimum time, in nanoseconds, between sending Ethernet frames after a transmission completes. 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-- Slot may not be specified for an ICA II system.

--ERROR-- Device state not on. Card slot = <slot_number>.

--ERROR-- Device already owned. Card slot = <slot_number>.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- Specified ETHERNET board is not on.

--ERROR-- Specified ETHERNET board is already in use.

--ERROR-- TRUNK <name> is the boot source. <Parameter> cannot be redefined.

--FATAL-- Not enough memory currently exists for required table space.

Remarks

- This command is required only in DIs configured with an Ethernet trunk. It is not required in DIs loaded over the Ethernet trunk. If this command is executed for the Ethernet trunk over which the DI boot was received, it redefines the trunk with the specified options. To redefine the Ethernet trunk over which the DI was booted, this command must be in the configuration file. Currently, the only options that may be redefined are TRUNK_NAME and MAX_FRAME_SIZE.
- To ensure correct service priority by the MPB in a DI, the ESCI board should be installed in a slot position greater than any of the boards in the DI except the MCI.
- A DI can support a maximum of three ESCI boards.

Examples

```
define_ether_trunk trunk_name=ETHER1,slot=4
```

DEFINE_EXCEPTION_SYSTEM (DEFES)
Exception List Command

Purpose Sets software version to load and load-time dump conditions for a specific DI. This software version and these dump conditions differ from those specified for other DIs on the DEFINE_BOOT_DEFAULTS command.

This command is valid only in an exception list. Do not use this command in a DI configuration procedure.

Format

DEFINE_EXCEPTION_SYSTEM
SYSTEM_ID = 0..0FFFFFF(16)
VERSION_NUMBER = 0..0FFFF(16)
SERVICE_SYSTEM = boolean
DELETE_DUMP_ON_ERROR = list 0..0FF(16)
ADD_DUMP_ON_ERROR = list 0..0FF(16)
TRANSMIT_RATE = 4800..2000000

Parameters

SYSTEM_ID (SI)

Lower six hexadecimal digits of the DI's 12-digit system ID. The upper six hexadecimal digits of the system ID are always 080025 and are supplied by the DEFINE_EXCEPTION_SYSTEM command processor.

VERSION_NUMBER (VN)

Software version number consisting of one through four hexadecimal digits. Any version number that starts with an alphabetic character must have a leading zero when it is entered in the exception list. Default is the DEFAULT_VERSION parameter value on DEFINE_BOOT_DEFAULTS.

SERVICE_SYSTEM (SS)

Whether or not the DI should be loaded. Values include TRUE, load DI; and FALSE, do not load DI. Default is TRUE.

NOTE

The Initialization Management Entity (ME) in a NOS/VE host or a DI uses SERVICE_SYSTEM when deciding to initialize (dump and/or load) DIs. SERVICE_SYSTEM does *not* control *where* Initialization ME gets files, such as boot files, configuration procedures, and TDPs. Do not use this parameter to control which host provides CDCNET file support. Instead, use the DEFINE_FILE_SUPPORT command.

DELETE_DUMP_ON_ERROR (DDOE)

List of error conditions to add to the list specified on DEFINE_BOOT_DEFAULTS. Under these conditions, the DI does not dump before loading. These error conditions are listed in the table of DI reset codes in the CDCNET Configuration Guide. Default is the list of error conditions specified on DEFINE_BOOT_DEFAULTS.

ADD_DUMP_ON_ERROR (ADOE)

List of error conditions causing the DI to dump memory before loading. This list may add conditions deleted by the DELETE_DUMP_ON_ERROR parameter on this command and on DEFINE_BOOT_DEFAULTS. These error conditions are listed in the CDCNET Configuration Guide. Default is to add no error conditions to the currently defined list.

TRANSMIT_RATE (TR)

Transmit rate (bits per second) at which initialization attempts to transfer load and dump information. It is used to slow down the initialization so it does not exceed the bandwidth of a constriction hidden in the load media. The effective rate is the lower of the specified transmit rate and available media bandwidth. Default is 2 million bits per second.

Remarks The SET_VERSION_LEVEL (SETVL) installation procedure (see NOS Installation handbook and CDCNET Installation section of the NOS/VE Software Release Bulletin) does not alter entries in the DEFINE_EXCEPTION_SYSTEM command. You must alter them directly on this command.

Examples The following command specifies that a DI with system ID 0800253000C2 is loaded with software version 0402. The DI does not dump when error codes 21 and 22 occur.

```
define_exception_system system_id=3000C2(16),..
version_number=0402(16),delete_dump_on_error=(21(16),22(16))
```

DEFINE_FILE_SUPPORT (DEFFS) (NOS Only) DI Configuration Procedure/NETOU Command

Purpose Defines the Independent File Access Management Entity (ME) in a DI and selects file types that can be accessed through the DI.

This command is optional in the system configuration procedures of MDIs loaded over an MCI that provide file support for the network. The host must be configured to run the Network File Server application, NETFS.

If this command is not in an MDI's configuration file, access to all CDCNET file types is supported. To have file support for a host whose MCI link was not used to load the MDI (that is, if multiple MCI boards are in the MDI or MTI), this command is required.

File support for each unique trunk name operates independently. That is, file support for multiple trunk names operates as if the trunk names were defined on separate DIs. Commands addressed to the Independent File Access ME for one trunk do not affect the Independent File Access ME 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)*

File types accessible through this DI. Files can be specified as a list of one or more of the following keyword values:

Keyword Value	Description
EXCEPTION	Exception list
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	CDCNET load procedures
ALL	All file types

Default is ALL.

TRUNK_NAME (TN)

Trunk name of the logical link used for the file access connection. If TRUNK_NAME is not specified, the default trunk is used. The default trunk is specified on DEFINE_SYSTEM. Default is the channel over which the DI was loaded. If the DI was not loaded over a channel and no default channel trunk was specified on DEFINE_SYSTEM, 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/MTI loaded across an MCI. The first time file support is redefined for the DI load media, you do not receive the following error message:

--ERROR-- File Support for trunk <name> is already defined.

The above error message appears during subsequent attempts to redefine file support. If this command is in the DI's configuration file, it redefines the default file type support of a DI loaded across an MCI. However, this command does not redefine file support if entered through the Network Operator Utility (NETOU). To redefine file support via NETOU, first cancel file support for all file types using CANCEL_FILE_SUPPORT. Then, redefine desired file types using DEFINE_FILE_SUPPORT.

Examples

```
define_file_support file_types=(exception,boot,library)

File Support is defined for trunk exception, boot, library.
```

DEFINE_HDLC_NET (DEFHN)
DI Configuration Procedure/NETOU Command

Purpose Configures a CDCNET HDLC network solution using a previously defined HDLC trunk, or several HDLC trunks. An "unable to start" error leaves the network defined but not started.

Format

DEFINE_HDLC_NET
TRUNK_NAME = list 1..32 of name
NETWORK_ID = 1..0FFFF(16)
NETWORK_NAME = name
COST = 0..7FFFFFFF(16)
RELAY_ALLOWED = boolean
ROUTING_INFO_NETWORK = boolean
START = boolean
ARCHITECTURE_TYPE = list 1..2 of keyword value
OUTPUT_QUEUE_LIMIT = 10000..500000
INCOMING_SERVICE_ACCESS = list 1..11 of keyword value
OUTGOING_SERVICE_ACCESS = list 1..11 of keyword value

Parameters

TRUNK_NAME (TN)

Logical name of the HDLC trunk or trunks used for the network solution. If more than one trunk name is specified, the logical name is a combination of all specified trunks. The trunk or trunks named by this parameter must be configured before this command executes.

NETWORK_ID (NI)

CDCNET network identifier of the HDLC 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_xxxxx*, where *xxxxx* is the network ID expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

Relative cost of the HDLC network solution as a path for routing data through the network. The cost of a network is calculated by dividing 100 million by the sum of the trunk data rates in bits per second. For example, the cost of a trunk with a speed of 56000 bits per second is 06FA(16). If the cost specified is greater than 0FFFF(16), 0FFFF(16) is used for for compatibility reasons. If the COST parameter is not specified, it is calculated by dividing 100 million by the sum of the speeds of the specified trunks. If the trunk speed specified on DEFINE_HDLC_TRUNK is 1200, default is 0FFFF(16).

If the COST parameter is specified, the cost of the specified network is static. If it is not specified, the cost is dynamic.

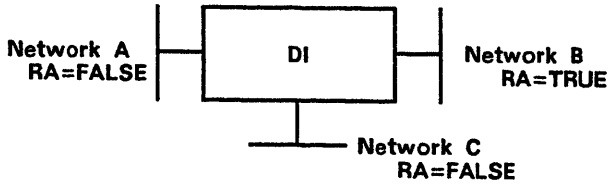
CDCNET routing uses cost to determine the least-cost routes for interconnecting networks. If multiple paths are available, the least-cost path is used. As a network becomes congested, its cost increases.

RELAY_ALLOWED (RA)

Whether or not the DI can use this network for relaying traffic through the DI. Possible values are TRUE, relay allowed; and FALSE, relay not allowed. Default is TRUE, relay allowed.

If RA is TRUE, the DI relays traffic between this network and other connected networks regardless of the definition of the other networks. If RA is FALSE, the DI still relays traffic between this network and any connected networks defined with RA=TRUE. If RA is FALSE, the DI system does not relay traffic between this network and other connected networks defined with RA=FALSE *unless* no other nonrestricted route exists between the networks.

The following example illustrates these rules.



Relay Between:	Allowed/Not Allowed:
Network A and Network B	Allowed (RA is TRUE for Network B).
Network A and Network C	Not allowed (RA is FALSE for both is FALSE for both Network A and Network C).
Network B and Network C	Allowed (RA is TRUE for Network B).

ROUTING_INFO_NETWORK (RIN)

Whether or not the network solution carries CDCNET routing information. Routing information pertains both to the proprietary CDCNET IS-IS routing protocol, as well as the ES-IS routing protocol. If RIN is TRUE, the DI sends routing information describing all the networks to which the DI connects over this network solution. If RIN is FALSE, the DI does not send routing information over the network solution. The DI appears unconnected to any network other than this network solution. Default is TRUE.

START (S)

Whether or not network starts after configuration. Default is TRUE.

ARCHITECTURE_TYPE (AT)

This parameter is no longer supported. Specification is valid but the value is ignored.

OUTPUT_QUEUE_LIMIT (OQL)

Maximum data, in bytes, retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. If messages must be discarded, newest output messages are discarded first.

Default depends on network cost (see COST parameter). If the cost is 6FA(16) or more, default is 30000 bytes. Otherwise, default is 60000 bytes.

INCOMING_SERVICE_ACCESS (ISA)

Specifies which services may be accessed coming in via the network solution. The keyword values are:

Keyword Value	Description
ALL	All services are allowed.
NONE	No services are allowed.
application	Any service not identifiable by keyword batch_support, interactive, file_transfer, or network_management.
batch_support	Any batch input and output services are allowed. This includes batch control facility, batch job submittal, and batch output.
interactive	All interactive services are allowed. Interactive service supports terminal user access. Specifying this keyword is the same as specifying all of the following interactive service keywords: nos_interactive, passthrough, t_passthrough, user_telnet, x25_terminal_gw, and ve_interactive.
nos_interactive	All interactive services provided by NOS mainframes are allowed.
passthrough	All site-defined passthrough services are allowed. A site-defined passthrough service is provided when a passthrough title is created during the execution of a Terminal User Procedure (TUP) when a line is activated. This service allows asynchronous interactive devices to make connections to passthrough titles of this type through asynchronous ports and to exchange information using CDCNET's transparent data feature.
t_passthrough	All user-defined passthrough services are allowed. A user-defined passthrough service is provided when an interactive terminal user creates a connection to the interactive passthrough server, and defines a passthrough title either by entering the required commands manually or by executing a Terminal User Procedure (TUP) via the DO command on the $CDCNET_COMMAND connection.

Keyword Value	Description
	This service allows asynchronous interactive devices to make connections to passthrough services of this type through asynchronous ports and exchange information using CDCNET's transparent data feature.
user_telnet	All TELNET services are allowed. User TELNET is an interactive service which provides CDCNET terminal users with access to the interactive services of remote hosts on a TCP/IP network.
x25_terminal_gw	All X.25 terminal services are allowed. The X.25 terminal gateway is an interactive service which provides CDCNET asynchronous terminal users with access to the interactive services of remote hosts on an X.25 network.
ve_interactive	All interactive services provided by NOS/VE mainframes are allowed.
file_transfer	All QTF/PTF file transfer services are allowed. File transfer services support user file (PTF) and queue file transfers (QTF).
network_management	All network alarm, clock, command, file, and log services are allowed.

Default is ALL.

OUTGOING_SERVICE_ACCESS (OSA)

Specifies which services may be accessed going out via the network solution. See the ISA parameter for the keyword value options.

Responses

HDLC network <network_name> defined for trunk <trunk_name>

HDLC network <network_name> defined for trunk(s) <trunk_name>, <trunk_name>...

HDLC network <network_name> defined and started for trunk <trunk_name>

HDLC network <network_name> defined and started for trunk(s) <trunk_name>, <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 <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.

--ERROR-- Different traffic_type values are configured for the trunks associated with the HDLC network solutions. All trunks associated with a single network solution must be configured with the same value for the traffic_type parameter.

--ERROR-- Trunk <trunk_name> is down. Unable to start network <network_name>.

--ERROR-- Trunk <trunk_name> is off. Unable to start network <network_name>.

--ERROR-- NETWORK <name> is the boot source. <Parameter> cannot be redefined.

--ERROR--Network <network_name> already defined for
trunk <trunk_name> by the boot source.
Network id is <network_id>.
The command network id is <network_id>.
Please reenter the command with correct network id.

--FATAL-- Not enough memory is currently available for required table space.

--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 Use multiple trunk networks instead of several single trunk networks when several HDLC trunks run between two DIs to reduce network overhead. Traffic is more equally distributed between multiple trunks.

When executed for the HDLC trunk over which the DI boot was received, this command redefines the network with the specified options. To redefine the HDLC network over which the DI was booted, this command must be in the configuration file.

Currently, redefinable options are as follows:

NETWORK_NAME
COST
RELAY_ALLOWED
ROUTING_ INFO_NETWORK
OUTPUT_QUEUE_LIMIT
INCOMING_SERVICE_ACCESS
OUTGOING_ SERVICE_ACCESS.

For an HDLC network which loads a DI to operate as a multiple-trunk network, that network solution must be redefined during configuration, specifying the associated trunks.

Examples In this example, an HDLC trunk links two separate sites (figure 6-2). Assume that interactive connections and QTF/PTF file transfers are permitted both ways between the two networks. But other services are not allowed. Here, you define service access with the DEFINE_HDLC_NET command.

Add the following to the configuration file for NDI_1:

```
define_hdlc_net..
   trunk_name=hdlc_trunk_1,network_name=name=hdlc_NET_1..
   network_id=3..
   incoming_service_access=(file_transfer,interactive)..
   outgoing_service_access=(file_transfer,interactive)
```

Add the following to the configuration file for NDI_21.

```
define_hdlc_net..
   trunk_name=hdlc_trunk_1,network_name=hdlc_net_1..
   network_id=3..
   incoming_service_access=(file_transfer,interactive)..
   outgoing_service_access=(file_transfer,interactive)
```

Note that only one side of the link needs to specify the service access restrictions. But defining restrictions on NDI_1 and NDI_21 prevents an operator on either side from changing the configuration to gain unauthorized access to services.

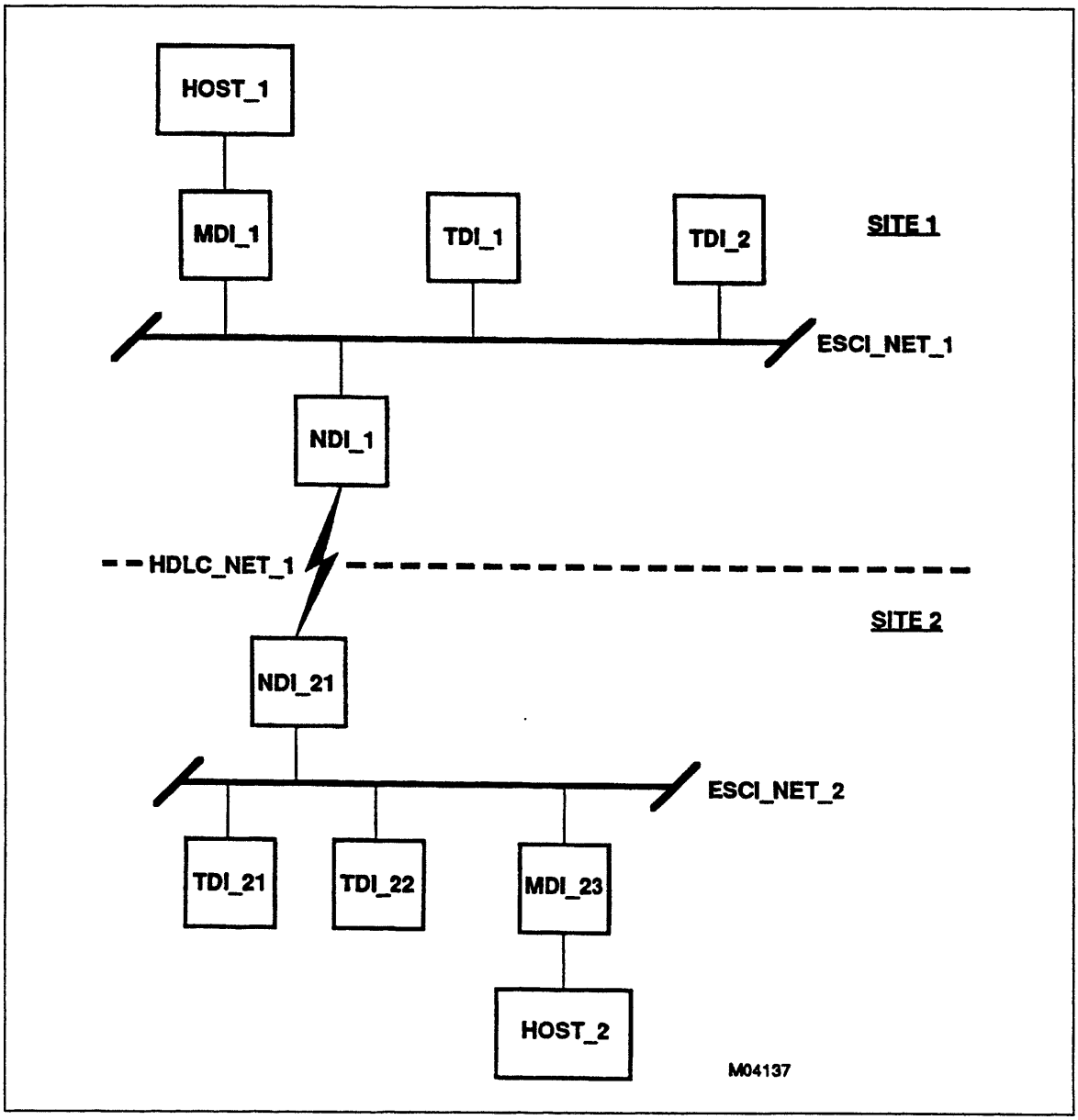


Figure 6-2. DEFHN Example

DEFINE_HDLC_TRUNK (DEFHT)
DI Configuration Procedure/NETOU Command

Purpose Configures an HDLC trunk.

Format

DEFINE_HDLC_TRUNK
LIM = 0..7
PORT = 0..3
TRUNK_NAME = name
LOCAL_ADDRESS = 1..255
REMOTE_ADDRESS = 1..255
OPTIONS = list of 1..6 of keyword value
MAX_UNACK_FRAMES = 0..127
SREJ_QUEUE_SIZE = 0..7
MAX_FRAME_SIZE = 132..32768
PF_RECOVERY_TIMER = 500..65535
ERROR_RECOVERY_TIMER = 500..65535
RETRANSMISSION_LIMIT = 1..65535
TRUNK_SPEED = 1200..256000
CLOCKING = keyword value
INTERACTIVE_BANDWIDTH = 1..9
DRIVER = keyword value
TRAFFIC_TYPE = keyword value
UNSTABLE_TRUNK_DISCONNECT_TIMER = 0..65535

Parameters

LIM (L)

LIM number for the port to which the HDLC trunk connects.

PORT (P)

Port number for the port to which the HDLC trunk connects.

TRUNK_NAME (TN)

Logical name of the HDLC trunk. The default name is constructed using the LIM and PORT parameters, as in $LIM1_PORT3.

LOCAL_ADDRESS (LA)

Address of the local HDLC station. LOCAL_ADDRESS is always 3 if the DI is loaded via the HDLC trunk.

REMOTE_ADDRESS (RA)

Address of the remote HDLC station. REMOTE_ADDRESS is always 1 if the DI is loaded via the HDLC trunk.

NOTE

The service DI (the DI providing support for loading, configuring, and dumping a Remote Terminal Interface (RTI)) must configure the HDLC trunk using 1 for the LOCAL_ADDRESS and 3 for the REMOTE_ADDRESS. Any other values for these addresses disable the reconnection of the RTI and the service DI, if the service DI is reset.

OPTIONS (O)

List of standard HDLC options the trunk supports. 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 transmission 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 unrecoverable error conditions by retransmitting the frame in error. Example error conditions include an invalid command, an information field exceeding 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 received next. Stations continually report these counts to each other during information exchange.
IFRAME_OFF	Does not include an IFRAME code in the HDLC control field.

Default HDLC options list is (REJ_ON, SREJ_ON, UI_ON, SIM_ON, RESET_ON, IFRAME_ON).

MAX_UNACK_FRAMES (MUF)

Maximum frames the local station can send without receiving an acknowledgement. The maximum number of frames that can be configured is 127. Default is 7.

SREJ_QUEUE_SIZE (SQS)

Size of the queue for holding frames received out of sequence and held by HDLC SSR pending receipt of missing frames whose transmission has been requested via the SREJ. Default is 7.

MAX_FRAME_SIZE (MFS)

Maximum frame size, in bytes, for a transmitted or received HDLC frame.

The value of this parameter must be coordinated with the value of the MAX_ DATAGRAM_SIZE parameter in the DEFINE_IP_NET command when MFS is used to configure an IP network over this HDLC trunk. Default is 1500.

PF_RECOVERY_TIMER (PRT)

P/F recovery timer value in milliseconds. This timer initiates P/F recovery when an acknowledgement is not received for an IFRAME in this time. Default is 500.

The following table shows best values for PF_RECOVERY_TIMER and ERROR_RECOVERY_TIMER (see following parameter description) values, based on trunk speed.

Trunk Speed	PRT (milliseconds)	ERT (milliseconds)
1200	32000	65535
2400	16000	65535
4800	8000	48000
9600	4000	24000
19200	2000	12000
38400	1000	6000
48000	750	4500
56000	500	3000
64000	500	3000
128000	500	3000
256000	500	3000

ERROR_RECOVERY_TIMER (ERT)

Error recovery timer value in milliseconds. This timer determines if P/F recovery has failed and initiates the next recovery level. Default is 3000.

The Network Configuration Utility (NETCU) uses the TRUNK_SPEED parameter value to determine P/F and error recovery timer values (see previous parameter description).

RETRANSMISSION_LIMIT (RL)

Maximum retransmissions allowed for a given control frame. Default is 5.

TRUNK_SPEED (TS)

HDLC trunk speed in bits per second. The DI uses trunk speed to generate data clocking for the trunk (except when clocking is EXTERNAL), and to configure the media with proper network cost and output queue limit values. NETCU uses trunk speed to set PF_RECOVERY_TIMER and ERROR_RECOVERY_TIMER parameter values (see PF_RECOVERY_TIMER parameter description). Values for this parameter include:

1200
2400
4800
9600
19200
38400
48000
56000
64000
128000
256000

Default is 56000. Specifying a trunk speed differing from the actual speed, even if you have specified EXTERNAL clocking, results in suboptimal performance. If clocking is external, trunk speed is used only to calculate network cost.

CLOCKING (C)

Whether the LIM internally generates the clock signal for data on this trunk or uses an externally generated clock signal. 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	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 RS-232 DD circuit for RS-232 ports or the RS-449 SR circuit for RS-449 ports. The EXTERNAL transmit data clock is derived from the RS-232 DB circuit for RS-232 ports or the RS-449 ST circuit for RS-449 ports.
TRANSMIT	LIM generates clocking for transmit data, but derives clocking for receive data from an external source. The transmit data clock matches the trunk speed specified for the trunk. The LIM supplies the transmit data clock on the RS-232 DA or RS-449 TT circuit. The LIM derives the receive data clock from the RS-232 DD or the RS-449 SR circuit.

Default 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.

For data clocking to work, LIMs supporting HDLC trunks must have the appropriate hardware configuration, and the CLOCKING parameter must be set.

INTERACTIVE_BANDWIDTH (IB)

Percentage of the trunk bandwidth used for transmitting data at interactive priority. Default is 7. For example, an interactive bandwidth value of of 7 results, on average, in 70 bytes of interactive priority data for every 30 bytes of batch priority data.

DRIVER (D)

Specifies the type of HDLC driver. Allowed keyword values include the following:

Keyword Value	Description
HDLC_NORMAL	Operates with other line types on the same CIM. It operates one HDLC line at 64 K bits per second and a combination of HDLC lines not exceeding 22,000 characters per second.
HDLC256	Operates one HDLC line at 128 K bits per second or 256 K bits per second on a dedicated CIM.
HDLC3x64	Operates one to three HDLC lines at speeds up to 64 K bits per second per line on a dedicated CIM.

Default is HDLC_NORMAL.

TRAFFIC_TYPE (TT)

Specifies the type of traffic used for this trunk. It also controls the use of the CDCNET proprietary Intranet header in data units transmitted on this trunk. Allowed keyword values include the following:

Keyword Value	Description
CDCNET	Used when the HDLC trunk connects two CDCNET systems. Each data unit transmitted includes the Intranet header. There is no restriction on the type of traffic which may be transmitted on this trunk.
NON_CDCNET_OSI	Used when the HDLC trunk connects a CDCNET system to a non-CDCNET OSI system. Data units transmitted do not include the Intranet header. This trunk can only be used to transmit traffic generated by the OSI network layer. Traffic received on this trunk is delivered at the OSI Network layer's Intranet SAP.
NON_CDCNET_IP	Used when the HDLC trunk connects a CDCNET system to a non-CDCNET TCP/IP system. Data units transmitted do not include the Intranet header. This trunk can only be used to transmit TCP/IP traffic. Traffic received on this trunk is delivered at the IP's Intranet SAP.

Default is CDCNET.

UNSTABLE_TRUNK_DISCONNECT_TIMER (UTDT)

Specifies the time, in seconds, that an unstable HDLC trunk is logically disconnected from a network solution. A trunk is considered unstable when it has gone into an inoperable state more than 3 times in the last 5 minutes, or 6 times in the last 30 minutes.

A value of 0 disables this feature. Default is 0.

Responses

HDLC trunk <trunk_name> defined.

--ERROR-- Trunk name <trunk_name> already defined.

--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.

--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.

--ERROR-- TRUNK <name> is the boot source. <Parameter> cannot be redefined.

--ERROR-- A line or trunk that is already defined is serviced by the same CIM as this trunk. The hdlc256 driver requires the CIM to be dedicated to this trunk.

--ERROR-- A line that is already defined is serviced by the same CIM as this trunk. The HDLC3X64 driver requires the CIM to be dedicated to HDLC or X.25 trunks.

--ERROR-- A trunk that is already defined for driver <driver> is serviced by the same CIM as this trunk. A CIM can only support one HDLC driver.

--ERROR-- Three HDLC or X.25 trunks that are already defined are serviced by the same CIM as this trunk. The hdlc3X64 driver cannot support more than three trunks.

--ERROR-- The DI was loaded via this HDLC trunk and the CIM switch setting specified the hdlc256 driver. The driver cannot be changed to hdlc_ normal or hdlc3x64 since these drivers do not handle high line speeds and communication could be lost.

--FATAL-- Not enough memory is currently available for required table space.

Remarks This command is required only in DIs configured with an HDLC trunk. This command is not required in systems booted over the HDLC trunk. If this command executes for the HDLC trunk over which the DI boot was received, it redefines the trunk with the specified options. To redefine the HDLC trunk over which the DI was booted, this command must be in the configuration file.

Currently, redefinable options are as follows:

TRUNK_NAME
PF_RECOVERY_TIMER
ERROR_RECOVERY_TIMER
RETRANSMISSION_LIMIT
TRUNK_SPEED
INTERACTIVE_BANDWIDTH

Examples

```
define_hdlc_trunk lim=5,port=0,local_address=1,remote_address=26

 HDLC trunk lim5_port0 defined.
```

DEFINE_INETD
INETD Configuration File Command

Purpose Defines the Internet Daemon (INETD) application.

This command is valid only in file $SYSTEM.TCP_IP.INETD_CONFIGURATION. Do not use this command in a DI configuration procedure.

Format

```
DEFINE_INETD
   SERVICE = name
   MANTA_FILE_NAME = file
```

Parameters

SERVICE (S)

Service name.

MANTA_FILE_NAME (MFN)

Name of the file used for communication between MANTA and INETD. File is created by INETD during initialization.

Examples

```
define_inetd ..
    service=inetd ..
    manta_file_name=$SYSTEM.TCP_IP.INETD_MANTA_SERVICE_FILE
```

DEFINE_IP_HOST (DEFIH)
DI Configuration Procedure/NETOU Command

Purpose

Configures an Internet Protocol (IP) host address and associated static routing information, such as the Ethernet system identifier and header format. DEFIH also configures IP host addresses for Address Resolution Protocol (ARP). If an IP host address is configured with an Ethernet address (system identifier), ARP does not send a request packet. If the IP host address is not configured, ARP broadcasts a request packet on the LAN to determine the Ethernet address. A configured IP host address routing entry never times out. However, the IP host address routing entry (learned and created by ARP) times out after 1 hour of inactivity.

DEFIH can be used to configure an ARP agent. If a DI is to respond to an ARP request for an IP address configured as an ARP agent, it responds with the configured Ethernet address.

An IP address must be configured for the following:

- The DI, one address for each directly connected IP network (IP_HOST=LOCAL).
- All CYBERs this DI supports (HOST_TYPE=CDC_HOST).
- Hosts or gateways that do not support ARP (HOST_TYPE=IP_HOST, IP_GW, IP_HOST_AGENT, or IP_GW_AGENT).

A DEFIH command is not required for:

- Any DI or host that does not use TCP/IP protocols.
- Any TCP/IP host that providing ARP services.

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)

IP address of the host/workstation/PC to be configured. The network number portion of the IP address must be previously defined by DEFINE_IP_NET. The format is similar to the decimal octet convention used by TCP/IP, except 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)

Host type associated with the configured address. The following types are allowed:

Keyword Value	Description
LOCAL (L)	The IP address refers to the DI you are configuring and is the default IP address for this DI. The IP address is the source address IP uses if upper layer protocols, such as TELNET, do not specify a source address. (There must be only one IP address of host type LOCAL configured for each IP network to which a DI physically connects.) If an ARP request is received for an IP address configured as LOCAL, the DI responds with the DI's Ethernet address. This IP address is used as the source IP address in ARP request and reply packets.
CDC_HOST (CH)	The IP address refers to an alternative address for this DI. That is, it is another IP address for this DI, besides the LOCAL host type IP address. A DI can have several IP addresses. Of these addresses, only one can be type LOCAL for each connected IP network; all others must be type CDC_HOST. For every CYBER this DI accesses and/or services, there should be a DEFIH with host type CDC_HOST. If an ARP request is received for an IP address configured as CDC_HOST, the DI responds with the DI's Ethernet address.

Keyword Value	Description
IP_HOST (IH)	The IP address refers to a host that is not the local DI. IP_HOST defines other IP hosts on an Ethernet network directly connected to the DI, so an IP address can map to a physical Ethernet system identifier, 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). • The IP network is not physically connected to the DI but is reached by a gateway. • ARP determines IP address/Ethernet address mapping.
IP_GW (IG)	The IP address refers to a gateway that is not the local DI. This type is like IP_HOST, except IP_GW further specifies the host is a gateway that can be a route to another IP network. If ARP is active, an IP gateway need not be defined.
IP_HOST_ AGENT (IHA)	The DI acts as an ARP agent for the configured IP address. When an ARP request packet is received for this IP address, the DI responds with the configured Ethernet address.
IP_GW_AGENT (IGA)	The DI acts as an ARP agent for the configured IP address. When an ARP request packet is received for this IP address, the DI responds with the configured Ethernet address. Requesting systems on a LAN use IP_GW_AGENT host type to reach remote networks. The DI supplies the Ethernet address to these systems.

Default is IP_HOST.

SYSTEM_ID (SI)

Ethernet address or CDNA system identifier of the host. This value 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)). You can omit this parameter for hosts not on Ethernet media or for host types LOCAL or CDC_HOST.

LAN_HEADER_FORMAT (LHF)

Local area network (LAN) header format used at the configured address. Ignored for host types LOCAL or CDC_HOST. The following values are allowed:

Keyword Value	Description
STANDARD_ HEADER (SH)	Host uses an IEEE 802.2 Ethernet header. This is the standard for CDCNET.
XNS_HEADER (XH)	Host uses a Xerox Networking Software (XNS) Version 2 Ethernet header. All older and many recent TCP/IP implementations use this header format.
IEEE_XNS_HEADER (IXH)	Host uses an IEEE 802.2 Ethernet header with a SNAP header to encapsulate XNS information. This is the new TCP/IP header format. However, most existing implementations use the older XNS_HEADER. Check 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 subnet <subnet_address> is not defined.

--ERROR-- IP address <ip_address> is invalid.

--ERROR-- System id is required for host_type of IP_HOST, IP_HOST_ AGENT, IP_GW, and IP_GW_AGENT.

--ERROR-- HOST_TYPE of LOCAL is already defined for 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)
DI Configuration Procedure/NETOU Command

Purpose

Configures an Internet Protocol (IP) network/subnet and associated routing information. An IP network/subnet address must be configured for every directly connected IP network/subnet, and for every indirectly connected IP network/subnet that cannot be learned via ARP, RIP, or EGP. If the IP network/subnet is directly connected, the physical network/subnet must be previously defined by the following commands: DEFINE_ETHER_TRUNK and DEFINE_ETHER_NET (for Ethernet networks), DEFINE_X25_TRUNK and DEFINE_X25_INTERFACE commands (for X.25 networks), or DEFINE_HDLC_TRUNK and DEFINE_HDLC_NET (for HDLC networks).

If at least one Ethernet trunk is configured as an IP network/subnet, ARP is loaded and configured into the system.

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..1500
TRUNK_NAME = name
ARP_LAN_HEADER_FORMAT = list 1..2 of keyword
SUBNET_MASK_SIZE = 10..30
OFFER_REVERSE_ARP_SERVICE = boolean
ENABLE_IP_BROADCAST = boolean

Parameters

IP_NETWORK (IN)

IP network number part of the IP address of the network to be configured. If IP_NETWORK is zero, any datagram to an IP network not in the routing tables is sent to the gateway specified by the IP_ADDRESS parameter. This is known as the *default route.*

The parameter format is similar to the decimal octet convention used by TCP/IP, except periods are replaced with commas and the list is enclosed in parentheses. One, two, or three values between commas must be specified for Class A, B, or C networks, respectively. If this network is subnetted, the specified values must be consistent with the subnet mask size. For example, IP network 192.2.53.0, which is not subnetted, is represented as (192,2,53,0) or (192,2,53). IP network 129.179.0.0, which is subnetted and has a subnet mask size of 24 bits, is represented as (129,179,10,0) or (129,179,10), where 10 is the subnet number.

IP_ADDRESS (IA)

IP address of the next gateway (hop) in the route to the destination IP network. To reach the destination IP network, this host must previously be configured by a DEFINE_IP_HOST command or learned by ARP. The format is similar to the decimal octet convention used by TCP/IP, except 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 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 (description follows) must also be specified. Default is 0.

MAX_DATAGRAM_SIZE (MDS)

Specifies the maximum datagram size, in bytes, that the IP network handles without fragmentation. If the hop count is nonzero, also consider the maximum size of intervening IP networks to avoid fragmentation. This applies only to directly connected networks.

The value of this parameter must be coordinated with the maximum frame size configured for the underlying trunk. The value must be smaller than the maximum frame size by at least 4 bytes.

Default is 576 bytes for DDN networks and 1500 bytes for Ethernet networks.

TRUNK_NAME (TN)

CDCNET trunk name of a directly connected network. The name must have been previously specified on DEFINE_ETHER_TRUNK, DEFINE_HDLC_TRUNK, or DEFINE_X25_TRUNK. If the value of the HOP_COUNT parameter is zero, this parameter is **required**; otherwise it is ignored.

For multiple trunk HDLC networks, any one trunk name from the list specified in the DEFINE_HDLC_NET command may be entered. The IP network/subnet utilizes all trunks defined for the associated HDLC network solution.

ARP_LAN_HEADER_FORMAT (ALHF)

Ethernet header format for ARP request packets. The following keyword values are allowed:

Keyword Value	Description
XNS_HEADER (XH)	ARP request packets use Xerox Networking Software (XNS) Version 2 Ethernet header. All older and many recent TCP/IP implementations use this header format.
IEEE_XNS_HEADER (IXH)	ARP request packets use IEEE 802.2 Ethernet header with a SNAP header. This is the new TCP/IP header format. However, most existing implementations use the older XNS_HEADER. Check vendor's documentation to determine which header is required.

Default is XNS_HEADER. If both header formats are specified, two ARP request packets are sent at the same time, one with an XNS header and the other with the 802.2 with SNAP header.

SUBNET_MASK_SIZE (SMS)

Whether or not the IP network is subnetted. Subnet mask size is the bits in the subnet mask. For subnetting to occur, the subnet mask must be at least 10, 18, or 26 bits for Class A, B, or C networks, respectively. The subnet mask size must be the same for all subnets of a given IP network. If the IP network is subnetted (by a previously issued DEFINE_IP_NET) and this parameter is omitted, the previously defined subnet mask size is used. Otherwise, the IP network defined by this command is not subnetted.

OFFER_REVERSE_ARP_SERVICE (ORAS)

Specifies whether or not the IP network is to be activated as a reverse ARP (RARP) Server, servicing RARP requests from all hosts in the directly connected network.

There must be at least one RARP Server defined for each LAN in order to provide RARP services. Since all RARP Servers defined through this command are of equal status, they all respond to RARP requests. To reduce the probability of collisions, do not define a large number of RARP Servers. Define at least two RARP Servers on each LAN to provide higher reliability, in case one RARP Server is down. Defining multiple RARP Servers implies entering DEFIN commands for multiple DIs on a single LAN.

ENABLE_IP_BROADCAST (EIB)

States whether or not the IP network is to receive broadcasts. If specified, broadcasts are sent onto the network from the device. Default is FALSE.

Responses

IP network <ip_network> is defined.

IP subnet <ip_network> is defined.

--WARNING-- The max_datagram_size must be at least 4 bytes smaller than the maximum frame size for the underlying trunk <trunk_name>. The specified value of <old_max_datagram_size> bytes was shortened to <new_max_datagram_size> bytes to meet this requirement.

--ERROR-- IP network <ip_network> is already defined.

--ERROR-- An IP network/subnet is already defined for X.25 trunk <trunk_name>.

--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-- Default gateway must be configured with non-zero hop count.

--ERROR-- IP address <ip_address> must be on a directly connected IP network.

--ERROR-- Trunk <trunk_name> is not defined as a network solution. A DEFINE_ETHER_NET command is required.

--ERROR-- Trunk <trunk_name> is not defined as a network solution. A DEFINE_HDLC_NET command is required.

--ERROR-- The X.25 Interface is not defined for trunk <trunk_name>. A DEFINE_X25_INTERFACE command is required.

--ERROR-- The max_datagram_size must be at least 4 bytes smaller than the maximum frame size for the underlying trunk <trunk_name>.

--ERROR-- An IP network is already defined for X.25 trunk <trunk_name>.

--ERROR-- IP subnet <ip_network> specified is too large for subnet mask size.

--ERROR-- IP subnet <ip_network> is invalid.

--ERROR-- IP subnet <ip_network> is already defined.

--ERROR-- IP network <ip_network> subnet mask size is inconsistent with the previously defined subnet mask size.

--ERROR-- IP subnet <ip_network> is not defined.

--ERROR-- The Reverse ARP Server can only be activated on an Ethernet trunk.

--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 ..
max_datagram_size = 576

IP Network 1.0.0.0 is defined.
```

DEFINE_IP_X25_PDN_HOST (DEFIXPH)
DI Configuration Procedure/NETOU Command

Purpose

Configures an IP host address and associated X.25 information when the host is located on an X.25 PDN. The command is logically equivalent to the DEFINE_IP_HOST command, which is used to configure hosts located on Ethernet, HDLC, and X.25 DDN trunks.

Format

DEFINE_IP_X25_PDN_HOST
IP_ADDRESS = list 4 of 0..255
REMOTE_DTE_ADDRESS = string 1..15
LOCAL_DTE_ADDRESS = string 1..15
HOST_TYPE = keyword value
ACCEPT_PDN_CHARGES = boolean
FACILITIES = string 2..62
INACTIVITY_TIMER = inter 1..65535 or *INFINITE*

Parameters

IP_ADDRESS (IA)

Specifies the IP address of the host/workstation/PC to be configured. The network number portion of the IP address must have previously been defined by a DEFINE_IP_NET command. 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).

REMOTE_DTE_ADDRESS (RDA)

Specifies the remote DTE address associated with the IP address. The address is specified as a string of digits 0 through 9. This is the DTE address used when creating a virtual circuit to transmit data to the associated IP address. This parameter is required if the HOST_TYPE is IP_HOST.

LOCAL_DTE_ADDRESS (LDA)

Specifies the local DTE address. The address is specified as a string of digits 0 through 9. This is the calling DTE address used when creating a virtual circuit to transmit data to the associated IP address. If the HOST_TYPE is LOCAL, this address is used as the default calling DTE address for all hosts on the network. Note that the LOCAL_DTE_ADDRESS and the REMOTE_DTE_ADDRESS are used by IP in determining when a new virtual circuit must be created. If the PDN being used does not automatically insert the calling DTE address into an X.25 call request, the local DTE address must be specified on the DEFINE_X25_INTERFACE command, or on a DEFINE_IP_X25_PDN_HOST command where HOST_TYPE is LOCAL, or on all DEFINE_IP_X25_PDN_HOST commands where the HOST_TYPE is IP_HOST.

HOST_TYPE (HT)

Specifies the type of host associated with the configured IP address. The default is IP_HOST. The following types are allowed:

Keyword Value	Description
LOCAL (L)	The specified IP address refers to this DI. This IP address is the source address used by IP if the upper layer protocols, such as TELNET, do not specify a source address. There must be only one IP address of type LOCAL configured for each IP network to which a DI is physically connected. When HOST_TYPE is LOCAL, the IP_ADDRESS and LOCAL_DTE_ ADDRESS are the only parameters that have meaning.
CDC_HOST (CH)	The specified IP address refers to the alternate address of 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 DEFINE_IP_HOST with HOST_TYPE of CDC_HOST for every CYBER that this DI is to access and/or service. When HOST_TYPE is CDC_HOST, the IP_ADDRESS and LOCAL_DTE_ ADDRESS are the only parameters that have meaning.
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 the PDN so that an IP address can be mapped to a DTE address and vice versa.

ACCEPT_PDN_CHARGES (APC)

Specifies how a call initiated from the remote DTE should be handled. If TRUE, an incoming call is accepted with either normal or reverse charging. If FALSE, incoming calls are accepted only with normal charging. The default is FALSE.

FACILITIES (F)

Specifies the facilities options as specified by the X.25 CCITT protocol. The parameter is a string containing an even number of hexadecimal digits. By default, no facilities are supplied in calls initiated to the remote DTE address.

INACTIVITY_TIMER (IT)

Specifies a timer, in minutes, for measuring inactivity on an X.25 virtual circuit. The virtual circuit is inactive when no data is transferred in either direction. If the inactivity timer expires, the virtual circuit is cleared. A new virtual circuit is established upon demand. The default is 10 minutes.

Responses

IP host address <ip_address> is defined.

--ERROR-- Facilities may only include digits "0".."9" and "A".."F".

--ERROR-- Facilities must have an even number of hexadecimal digits.

--ERROR-- Hosts on an ETHERNET network must be defined with the DEFIH command.

--ERROR-- Hosts on a DDN network must be defined with the DEFIH command.

--ERROR-- Host_type of LOCAL is already defined for network <network_address>.

--ERROR-- IP address <ip_address> is invalid.

--ERROR-- IP host address <ip_address> is already defined.

--ERROR-- IP network <network_address> is not defined.

--ERROR-- IP subnet <subnet_address> is not defined.

--ERROR-- Local_dte_address may only include digits "0".."9".

--FATAL-- Not enough memory is currently available for required table space.

--ERROR-- Remote_dte_address must be specified for host_type IP_HOST.

--ERROR-- Remote_dte_address may only include digits "0".."9".

DEFINE_LINE (DEFL)
DI Configuration Procedure/NETOU Command

Purpose

Defines logical line name, physical hardware address, the TIP name servicing the line, physical line attributes, and connect timeout values of a terminal communication line or URI parallel interface line.

Examples at the end of this description show DEFINE_LINE entered two ways: in a configuration procedure and through the Network Operator Utility (NETOU) while the network is running.

NOTE

For lines supported by X.25 Async TIPs, do not use this command to define the lines. Instead, use DEFINE_X25_ASYNCTIP, described later in this chapter.

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
VALIDATE_USERS = boolean

Parameters

LIM (L)

Slot number for the Line Interface Module (LIM) or Unit Record Interface (URI) board in the MTI/TDI to which the line connects. An MTI or TDI allows for up to eight LIMs/URIs.

PORT (P)

LIM port number that connects to the line. The ports supported per LIM is LIM model-specific. Depending on the LIM model supporting the line, this parameter range is 0 through 1, 0 through 3, or 0 through 7.

TIP_NAME (TN)

Name of the TIP servicing the line. See DEFINE_TIP description (parameter TIP_NAME or USER_TIP_NAME for user-defined TIPs) for allowed TIP names.

NOTE

For TIPs supporting X.PC, if TDPs configure the terminal devices (see DEFINE_TERMINAL_DEVICE), the TDP must not contain a DEFTD command with a nonzero device address. The X.PC protocol starts only when the device address is zero.

LINE_NAME (LN)

Logical name of the line or URI parallel interface. The default line name is constructed from the SLOT and PORT parameter values, as in $LIM3_PORT1 or $URI2.

If TIP_NAME is ASYNCTIP, HASP, NJEF, or MODE4, and a user connects to a NOS host, the NOS terminal name is based on the line name (unless a TDP containing a DEFINE_TERMINAL_DEVICE naming the terminal is also specified). Ensure the terminal name for each line is unique throughout the network.

LINE_TYPE (LT)

Line type, which is either SWITCHED or DEDICATED. Default is SWITCHED. For URI-supported lines, the value for this parameter is fixed as DEDICATED.

The TIP servicing a line handles switched and dedicated lines the same, except for the defaults for CONNECTION_CONNECT_TIMEOUT (CCT) and CONNECTION_DISCONNECT_TIMEOUT (CDT) parameters. For dedicated lines, default for CCT and CDT is INFINITE. For switched lines, default for CCT and CDT is 120.

LINE_SUB_TYPE (LST)

Subtype of line, which a site can use to further qualify the line type, such as WATS or INWATS. The line subtype is reported in all device accounting messages generated for the line.

CARRIER_TYPE (CT)

Type of carrier control on the line, which is either CONSTANT or CONTROLLED. Default is CONSTANT. Ignored if TIP_NAME=URITIP.

LINE_SPEED (LS)

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 following line speed ranges are supported by individual TIPs:

TIP	Line Speed Range
ASYNC	50 through 38400
HASP NJEF NTF	1200 through 64000
BSC3270	50 through 9600.
MODE4	1200 through 19200.

This parameter is only informational either if the LIM board does not generate data clocking for the line or if auto recognition of line speed is requested for an asynchronous line.

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.
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.
SCPNE	Auto recognition of line speed, code set, and parity. Parity types are none and even.

Default is NONE.

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.

TRANSMISSION_BLOCK_SIZE (TBS)

Defines the transmission block size to be used for transmission blocks exchanged with the device(s) on this line.

This value reserves input buffer space in the CIM. Use the maximum block size sent to CDCNET by a HASP, Mode 4, BSC 3270, NJEF, or NTF connection. Otherwise, CDCNET may disconnect the line because of excessive errors on the 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: ASYNC, HASP, MODE4, NJEF, BSC3270, and NTF.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines the amount of time the line user has to create the first $INPUT/$OUTPUT connection. You can either specify an amount of time within the range of 20 through 1000 seconds, or use the keyword value INFINITE, indicating an infinite period of time. If no connection is established within that time, the line is disconnected. It may be necessary to increase this value if the VALIDATE_USERS parameter is TRUE.

The value of 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. The timer has a precision of +/- 2 seconds.

The default is 120 seconds for a switched line and INFINITE for a dedicated line.

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 is 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. You can either specify an amount of time within the range of 0 through 1000 seconds, or use the keyword value INFINITE, indicating an infinite period of time. The default is 120 seconds for a switched line and INFINITE for a dedicated line. The value of 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. This parameter is ignored if TIP_NAME=URITIP.

For lines supporting the X.PC protocol, 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 or not the Clear to Send and Request to Send flow control are used to stop and resume the flow of input and output data. The options are ON and OFF. Default is OFF. All synchronous TIPs ignore this parameter. 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). EXTERNAL receive data clock is derived from the RS-232 DD circuit for RS-232 ports or the RS-449 SR circuit for RS-449 ports. EXTERNAL transmit data clock is derived from the RS-232 DB circuit for RS-232 ports or the RS-449 ST circuit for RS-449 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 RS-232 DA or RS-449 TT circuit.
RECEIVE	Specifies that the LIM generates the clocking for received data, but derives the clocking for transmitted data from an external source. The receive data clock matches the line speed specified for the line. The LIM supplies the receive data clock on the RS-232 DD or RS-449 SR circuit. The LIM derives the transmit data clock from the RS-232 DB or the RS-449 ST circuit.

Keyword Value	Description
TRANSMIT	Specifies that the LIM generates clocking for transmit data but derives 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 RS-232 DA or RS-449 TT circuit. The LIM derives the receive data clock from the RS-232 DD or the RS-449 SR circuit.

Default is INTERNAL.

Set Clocking 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 generate the transmit clock as defined by the RS-232 standard. When using a modem, CLOCKING must be set to EXTERNAL and modem cable TN108 must be used, since the modem generates both clocking signals. Inappropriate selection of INTERNAL clocking causes 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 0.
MARK	The parity bit is always 1.
EVEN	The parity bit is set so that the sum of the parity and data bits is even.
ODD	The parity bit is set so that the sum of the parity and data bits is odd.
NONE	The parity bit is considered a data bit. This type of parity is significant only during transparent input or output. When input or output mode is not transparent, parity is treated as if it were ZERO.

Default data parity is EVEN.

All values are allowed for the ASYNC TIP. This parameter is ignored by the URI, BSC3270, HASP, and MODE4 TIPs.

VALIDATE_USERS (VU)

Specifies if users of this line are validated when the validation is enabled by the DEFINE_USER_VALIDATION command. If TRUE is specified, validation is enabled. Default is the VU value for the TIP associated with this line.

Responses

Line <xxx> definition in progress.

--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-- 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.

--ERROR-- LIM <xx>, port <yy> not responding or not installed.

--ERROR-- TIP <name> is not a CDC defined TIP name.

--ERROR-- TIP <name> not defined.

--ERROR-- TIP type <name> is not supported on the 8-port LIM.

--ERROR-- Line <name> previously defined.

--ERROR-- Line speed <line_speed> is not supported by the specified TIP.

--ERROR-- Load module <TIP name>_CIM is not available.

--ERROR-- An HDLC or X.25 trunk that is already defined is serviced by the same CIM as this line. The HDLC driver for the trunk requires the CIM to be dedicated to HDLC or X.25 trunks.

--FATAL-- Not enough memory is currently available for required table space.

--FATAL-- No CIM RAM space for I/O processor <xx> in board slot <xx>.

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 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.

For a dedicated URI printer line, a power-off condition causes log messages to occur (about every 2 minutes). If a printer is powered off for long periods of time, you should stop its line with a STOP_LINE command (see CDCNET Network Operations manual).

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_procedure=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 is not redefined. For the line to be redefined every time the TDI's software is reloaded, the DEFINE_LINE command has to be placed in the TDI's configuration procedure.

```
nou/send_command command='define_line line_name=line23,..
lim=2,port=3,tip_name=asynctip,line_speed=9600',system=tdi_84
```

DEFINE_LLC_2_TRUNK (DEFL2T)
DI Configuration Procedure/NETOU Command

Purpose

Defines LLC 2 parameters and X.25 parameters applicable only when running X.25 over Ethernet. When the peer is a CDCNET device, this command is normally not needed. As this command (as well as DEFINE_X25_INTERFACE) is simulated internally in the CDCNET device, the defaults described below are automatically used. However, when the peer is a non-CDCNET device, this command (as well as DEFINE_X25_INTERFACE) is needed, especially in defining the X.25 virtual circuit call establishment method by the mode parameter.

The LLC 2 trunk is considered as an X.25 trunk by several other commands. This includes all of the X25 interface commands (such as DEFINE_X25_INTERFACE) and CANCEL_X25_TRUNK. However, for CONS static routing, the ADD_NON_CDC_ROUTING_ENTRY command should define a static routing entry based on the ETHER trunk, not the LLC 2 trunk.

This command is required to configure any X.25 user other than CONS to run on top of X.25 over Ethernet. An X.25 user on top of an LLC 2 trunk is the same an X.25 user on top of an X.25 trunk except that the LLC 2 trunk does not support PVCs.

NOTE

You must configure CONS, even if CONS is not a user of the trunk, whenever you use DEFL2T to manage the LLC 2 services associated with the trunk.

Format

DEFINE_LLC_2_TRUNK
ETHER_TRUNK_NAME = name
PEER_MAC_ADDRESS = string(12)
PEER_LSAP_ADDRESS = 1..255
TRUNK_NAME = name
MODE = keyword value
RECEIVE_WINDOW_SIZE = 1..127
TRANSMIT_WINDOW_SIZE = 1..127
ACKNOWLEDGEMENT_TIMER = 500..65535
POLL_BIT_TIMER = 500..65535
REJECT_TIMER = 500..65535
BUSY_STATE_TIMER = 500..65535

Parameters

ETHER_TRUNK_NAME (ETN)

Specifies the logical name of the Ethernet trunk to be used for the LLC 2 trunk. The Ethernet trunk with this name must be configured prior to the execution of this command.

PEER_MAC_ADDRESS (PMA)

Specifies the MAC address of the peer system.

PEER_LSAP_ADDRESS (PLA)

Specifies the LSAP address associated with the X.25 packet-level entity of the peer system. The most common values used are 7E(16) and FE(16). The default is FE(16), that is 254 (10).

TRUNK_NAME (TN)

Specifies the logical name of the LLC 2 trunk. The default name is constructed from the 12-digit peer MAC address as follows: $X25_LLC_2_<peer MAC address>.

MODE (M)

Specifies the mode of operation of the X.25 trunk. Allowed keyword values include the following:

Keyword Value	Description
DCE	The DI operates as the Data Communication Equipment end for the X.25 trunk.
DTE	The DI operates as the Data Terminating Equipment end for the X.25 trunk.
RESTART	The DI initially operates as a DTE, but may switch to DCE-base on X.25 RESTART packet negotiation.
DYNAMIC	The DI operates as a DTE but uses the X.25 Reference Number Facility to dynamically assign logical channels during virtual circuit establishment.

The default mode is DYNAMIC.

RECEIVE_WINDOW_SIZE (RWS)

Specifies the maximum number of LLC 2 I-frames the peer LLC 2 entity can send without receiving an acknowledgement. The default is 16.

TRANSMIT_WINDOW_SIZE (TWS)

Specifies the maximum number of LLC 2 I-frames the local LLC 2 entity can send without receiving an acknowledgement. The default is 16.

ACKNOWLEDGEMENT_TIMER (AT)

Specifies the value of the timer, in milliseconds. The LLC 2 timer defines the time interval during which the local LLC 2 entity expects to receive either an acknowledgement to one or more I-frames or a response PDU to a sent unnumbered command PDU. The default is 3000.

POLL_BIT_TIMER (PBT)

Specifies the value of the timer, in milliseconds. The LLC 2 timer defines the time interval during which the local LLC 2 entity expects to receive a PDU with the final bit set in a response to a sent LLC 2 command PDU with the poll bit set. The default is 3000.

REJECT_TIMER (RT)

Specifies the value of the timer, in milliseconds. The LLC 2 timer defines the time interval during which the local LLC 2 entity expects a response to a sent Reject (REJ) PDU. The default is 5000.

BUSY_STATE_TIMER (BST)

Specifies the value of the timer, in milliseconds. The LLC 2 timer defines the time interval during which the local LLC 2 entity waits for an indication of the clearance of a busy condition from the peer LLC 2 entity. The default is 10000.

Remarks

- This command is useful whenever you want to use some capability of a non-CDCNET device that communicates to a PDN over X.25. For example, when you want to use a router that has a higher HDLC speed over X.25 than over a CDCNET DI. This command allows CDCNET to talk to the router.
- This command is needed to support the X.500 server application on an EP/IX system when it is providing services out a CDCNET X.25 trunk.

Responses

LLC 2 trunk <trunk_name> defined for trunk <ether_trunk_name>.

--ERROR-- Trunk <ether_trunk_name> is not defined.

--ERROR-- Trunk <ether_trunk_name> is not an ETHERNET trunk.

--ERROR-- Trunk name <trunk_name> already defined.

--ERROR-- Peer MAC address <peer_mac_address> already defined for trunk <ether_trunk_name>.

--ERROR-- Peer MAC address <peer_mac_address> contains non-hexadecimal characters.

--FATAL-- Not enough memory is currently available for required table space.

DEFINE_NAME_RESOLVER (DEFNR)
DI Configuration Procedure/NETOU Command

Purpose Defines and starts the DI Domain Name Resolver (DNR). Also, it verifies that User Data Protocol (UDP) has been configured in the Gateway Device Interface (GDI). If UDP has been configured, the DNR is successfully initialized and requests can be accepted and queries sent out to Domain Name Servers (DNS).

Format

DEFINE_NAME_RESOLVER
DOMAIN_NAME = string 1..255
SERVER_ADDRESSES = list 1..4, 4 of integer 0..255
RETRY_LIMIT = integer 0..256
RETRY_TIMER = integer 1..32

Parameters

DOMAIN_NAME (DN)

Defines the name of the local administrative domain for the DNR. When requested to obtain information about a name and the name is a simple domain label, the DNR appends the DOMAIN_NAME parameter to the label before querying a DNS. The string consists of a series of domain labels separated by periods (for example, arh.cdc.com). A domain label in the string can be up to 63 characters long.

SERVER_ADDRESSES (SA)

Defines the order in which to query the IP addresses of the DNSs. The IP addresses are used in the order in which they are entered to communicate with the DNSs. At least one address is required which becomes the primary DNS. Up to three additional addresses may be entered specifying the secondary DNS.

RETRY_LIMIT (RL)

Defines the maximum number of times a query is reissued over UDP. Default is 4.

RETRY_TIMER (RT)

Defines the timeout value used for UDP datagrams. It is used to reissue a UDP datagram if no response is received within the RETRY_TIMER period. Also, it is the value used to timeout a request if the application has not specified a timeout value. Default is 10 seconds.

Responses The TCP/IP Name Resolver is defined.

--ERROR-- The TCP/IP Name Resolver is already defined.

--ERROR-- Domain_name cannot contain the character <character>. Domain_name can only contain the characters A-Z, a-z, 0-9, hyphen (-) and underscore (_).

--ERROR-- A domain_name label is too long. Each label cannot be greater than 63 characters.

--ERROR-- UDP is not defined.

Examples

```
define_name_resolver domain_name = 'arh.cdc.com' ..
server_address = ((129,179,60,48) ..  (GRAY)
(129,179,52,203) ..  (ROYAL)
(129,179,48,30)) ..  (ODEON)
retry_limit = 5 ..
retry_timer = 10

The TCP/IP Name Resolver is defined.
```

DEFINE_NP_GW (DEFNG) (NOS Only)
DI Configuration Procedure/NETOU Command

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. Control Data does not recommend using this title format. The format of this second title type is *$GW_NP_xx*, where *xx* is the ASCII representation of a two-digit hexadecimal number, which is specified by the NOS_DHOST_NUMBER parameter. Only the *xx* portion of the title is site-definable; the *$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. See the OUTCALL command in the NAM Network Definition Language manual.
- 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 manual.

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..255
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. Default is TRUNK_NAME.

TRUNK_NAME (TN)

Specifies the trunk name of the NOS host/MDI logical link that is to be used to support A-to-A traffic with this host. Default is the trunk specified on the DEFINE_SYSTEM command. Default on that command is the channel over which the MDI/MTI was loaded.

NOS_PROTOCOL_ID (NPI)

Specifies one or more protocol IDs that identify outcalls to be routed to NOS systems by host node number. Defaults 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. Default 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 *$GW_NP_xx*, where the *xx* portion consists of the ASCII equivalent of the hexadecimal digits. A default title is constructed from the coupler node number received from the host when the gateway's connection to the host is opened. That is, the default title is *$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. No default.

DEFAULT_TRANSLATION_DOMAIN (DTD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the catenet to which the services of this gateway are to be made available. Default is CATENET.

DEFAULT_SEARCH_DOMAIN (DSD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the catenet that should be searched in order to find the service that corresponds to the title information received by the gateway in ICN/AP/R messages. Default is CATENET.

START (S)

Specifies whether or not the NP gateway should be started after it is configured. Default is TRUE; started. Currently, the START parameter is ignored. Its value is always 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)

NP gateway trunk_name is defined.
```

DEFINE_NP_INTERFACE (DEFNI) (NOS Only) DI Configuration Procedure/NETOU Command

Purpose

Defines the network block protocol interface program (BIP) to a 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 needed only in NOS MDIs with more than one MCI, or when you choose to have an MDI 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 need to be specified in a system configuration procedure only if the host is installed with a version of software (before NOS version 2.5.2) that 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..255
   COUPLER_NODE = 0..255
   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 before executing this command.

MDI_NODE (MN)

MDI node identifier of the logical link. If specified, 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. No default.

COUPLER_NODE (CN)

Coupler node identifier of the logical link. If specified, this parameter must be set equal to the ND parameter on the NOS host's EST definition for this logical link. Default is the coupler node number obtained from the host.

INTERFACE_NAME (IN)

Logical name of the NP interface. Default is the trunk name.

CONGESTED_THRESHOLD (CT)

Specifies the number of messages in BIP's outgoing queue at which the network products interface is considered congested. Default 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. Default 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 <integer> 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

NP interface $mci3 is defined.
```

DEFINE_NP_TERMINAL_GW (DEFNTG) (NOS Only) DI Configuration Procedure/NETOU Command

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-to-A) interactive connection or a terminal-to-Remote Batch Facility (T-to-RBF) connection. These titles are specified as 1- to 31-character logical identifiers of the NOS host.

This command also defines the trunk between a NOS host and the MDI to be used to support T-to-A interactive and T-to-RBF batch traffic.

Format

DEFINE_NP_TERMINAL_GW
TITLE = list 1..15 of name(s)
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 = name
DEFAULT_BATCH_TERMINAL_CLASS = keyword value
START = boolean

Parameters

TITLE or TITLES (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. Default is TRUNK_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-to-A traffic or batch T-to-RBF traffic with this host. Default is the trunk specified on the DEFINE_SYSTEM command. Default on that command is the channel over which the MDI/MTI was loaded. If the MDI/MTI was not loaded over a channel, and no default channel trunk is specified on the DEFINE_SYSTEM command, then no default channel trunk exists.

TRANSLATION_DOMAIN (TD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the catenet to which the services of this gateway are to be made available. Default 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. (See the TERMINAL_MODEL_MAPPING parameter description described next.) Default 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 a terminal connection request (ICN/TE/R) message sent to NAM. A default terminal model mapping table is provided by the NP terminal gateway, shown in the following table.

If a terminal model in the default mapping table is redefined by a DEFNTG command, the DEFNTG defined pairing replaces the pairing from the default table. A DEFNTG definition of a new terminal model adds the new pairing to the terminal model mapping table maintained by the NP terminal gateway.

If no terminal model is specified by a user or if a specified terminal model is not found in the following table, the default terminal class is used. (See DEFAULT_TERMINAL_CLASS parameter).

Terminal Model	Class	Manufacturer
cdc_721	3	CDC
cdc721	3	CDC
cdc_722	2	CDC
cdc722	2	CDC
cdc722_30	7	CDC
cdc_722_30	7	CDC
cdc_200UT	10	CDC
cdc_714_30	11	CDC
cdc_711	12	CDC
cdc_714	13	CDC
cdc_734	15	CDC
dec_vt100	7	Digital Equipment Corp.
dec_vt100_gold	7	Digital Equipment Corp.
dec_vt220	7	Digital Equipment Corp.
ibm_hasp_post	9	IBM (HASP postprint)
ibm_hasp_pre	14	IBM (HASP preprint)
ibm_3270	18	IBM model 2
ibm_3270_2	18	IBM model 2 (24 x 80)
ibm_3270_3	18	IBM model 3 (32 x 80)
ibm_3270_4	18	IBM model 4 (43 x 80)
ibm_3270_5	18	IBM model 5 (27 x 132)
mac_connect_10	7	Macintosh/Connect 1.0
mac_connect_11	7	Macintosh/Connect 1.1
mac_connect_20	7	Macintosh/Connect 2.0
mode4	15	CDC
mode4C30	11	CDC
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
pc_connect_13	7	IBM PC/Connect 1.3
sun_160	7	Sun Microsystems
tek_4109	8	Tektronix
tek_4115	8	Tektronix
tek_4125	8	Tektronix
vt100	7	Digital Equipment Corp.
z19	7	Zenith
z29	7	Zenith
zen_z19	7	Zenith
zen_z29	7	Zenith

BATCH_TITLE (BT)

Specifies the title by which the host associated with this MDI is to be known by Remote Batch Facility (RBF) users accessing the host through CDCNET and/or the name of the control facility for stand alone printers connected to NOS. This title must be unique. If omitted, batch access is not supported by this gateway definition.

DEFAULT_BATCH_TERMINAL_CLASS (DBTC)

Specifies the default terminal class for batch devices. Allowed terminal classes and batch device types associated with each class are as follows:

Class	Terminal Model	Batch Device Type
9	ibm_hasp_post	IBM HASP postprint
10	cdc_200ut	CDC 200UT Mode 4A
11	cdc_714_30 or mode4c30	CDC 714-30 Mode 4C
12	cdc_711	CDC 711-10 Mode 4C
13	cdc_714	CDC 714-10/20 Mode 4C
14	ibm_hasp_pre	IBM HASP preprint
15	cdc_734 or mode4	CDC 731/734 Mode4A
18	ibm_3270	IBM 3270

Default is 9.

START (S)

Specifies whether or not the NP terminal gateway should be started. Default is TRUE; started. Currently, the START parameter is ignored. Its value is always 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-- Batch title <name> is the same as an interactive title specified for this gateway. The batch title must be unique.

--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 18.

--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

 NP terminal gateway ARHSES is defined.
```

DEFINE_OPERATOR_SUPPORT (DEFOS) (NOS Only) DI Configuration Procedure/NETOU Command

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 do 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 that is to be used for the operator support connection. Default is the trunk specified on the DEFINE_SYSTEM command. Default on that command is the channel over which the MDI/MTI was loaded. If the MDI/MTI was not loaded over a channel, and no default channel trunk was specified on 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 NETOU.

Examples

```
define_operator_support

Operator Support is defined for trunk $mci2.
```

DEFINE_OUTCALL_GATEWAY (DEFOG)
DI Configuration/NETOU Command

Purpose Configures the Outcall Gateway. The Outcall Gateway must be installed into every DI having devices configured as outcall servers. See the CDCNET Terminal Interface manual for detailed information.

Format

DEFINE_OUTCALL_GATEWAY
REGISTRATION_TITLE = list 1..2 of name
DEFAULT_INACTIVITY_TIMER = 120..14400 or *INFINITE*
START = boolean

Parameters

REGISTRATION_TITLE (RT)

Specifies one or two additional titles that may be used for server registration. To become a server, a device must connect to the Outcall Gateway and register itself as a server. The address to which to make this connection is obtained by completing a title translation to the $OUTCALL title or to one of the titles specified by this parameter.

The title $OUTCALL is always registered even if this parameter is not specified.

DEFAULT_INACTIVITY_TIMER (DIT)

Specifies the maximum time, in seconds, that an outcall connection can remain idle (no data has been transferred in either direction on the connection). This value establishes a default timer for all server devices. A continuous period of inactivity that exceeds this value results in termination of the connection, making the server available to a subsequent client.

The default value, INFINITE, specifies that outcall connections are not to be timed out. The default can be overridden for individual servers by using the INACTIVITY_TIMER parameter of the DEFINE_SERVER_DEVICE command (see the Terminal Interface manual).

START (S)

Specifies whether or not the defined service is to be started. Default is YES.

Responses

Outcall Gateway is defined.

Outcall Gateway is defined and started.

--ERROR-- Outcall Gateway is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_outcall_gateway dit=120 s=no

Outcall Gateway is defined.
```

DEFINE_PASSTHROUGH_SERVICE (DEFPS)
DI Configuration Procedure/NETOU Command

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.

NOTE

In future releases, this command will no longer be supported. You should start using the DEFINE_OUTCALL_GATEWAY command as soon as possible.

Format

DEFINE_PASSTHROUGH_SERVICE
TITLE = name
INACTIVITY_TIMER = 120..14400 or *INFINITE*
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. Default 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. Default is INFINITE; passthrough connections are not to be timed out.

START (S)

Specifies whether or not the defined service is to be started. Default 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.

You can set the INACTIVITY_TIMER at this level instead of at the gateway level.

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

Passthrough service termpass defined.
```

DEFINE_PRINTER_MODEL_ATTRIBUTES (DEFPMA) DI Configuration Procedure/NETOU Command

Purpose Defines the printer attributes 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 the batch device is a particular printer model, having that model's attributes. DEFPMA can be specified in a DI configuration procedure or via NETOU.

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 = 2..12
VFU_TOP_FORM = boolean
MAXIMUM_VFU_LENGTH = 0..255
INITIALIZATION_PROCEDURE = name
INITIALIZATION_SEQUENCE = list 1..31 of <ccode>
EIGHT_LPI_SEQUENCE = list 1..7 of <ccode>
FILE_PREFIX_PROCEDURE = name
FILE_PREFIX_SEQUENCE = list 1..31 of <ccode>
FILE_SUFFIX_SEQUENCE = list 1..31 of <ccode>
SIX_LPI_SEQUENCE = list 1..7 of <ccode>

NOTE

- For character code (ccode) values, see the CDCNET Configuration Guide.
- If you cannot specify all desired parameter values on one DEFINE_PRINTER_MODEL_ATTRIBUTES command, use one or more CHANGE_PRINTER_MODEL_ATTRIBUTES commands to specify additional attributes.

Parameters **TERMINAL_MODEL (TM)**

Name of the printer terminal model (1 through 25 characters long) for which attributes are being defined.

If the name begins with "POSTSCRIPT_", CDCNET does the following special processing:

- Substitutes the file page width for NOS/VE or device page width for NOS for any occurrence of the following string in the device's file prefix procedure:

  $PW

- Substitutes the device forms size, as determined by the DEFINE_BATCH_DEVICE FORMS_SIZE parameter for NOS/VE and the PSU PRSIZE parameter for NOS, for any occurrence of the following string in the device's file prefix procedure:

  $FS

- Inserts the "\" character in front of each of the following characters when encountered in nontransparent output file data:

  \
  (
)

This parameter must not be the same as any other terminal model already defined by Control Data or by your site.

AUTO_PAGE_EJECT_CHANNEL (APEC)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Channel causing the printer to skip automatically to the next top-of-form channel. May not be the same as the BOTTOM_OF_FORM_CHANNEL (BOFC) value. Default is 2.

CHANNEL_1_SEQUENCE (C1S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "8" or "H" format effector is recognized in output lines. No default.

CHANNEL_2_SEQUENCE (C2S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "7" or "G" format effector is recognized in output lines. No default.

CHANNEL_3_SEQUENCE (C3S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "6" or "F" format effector is recognized in output lines. No default.

CHANNEL_4_SEQUENCE (C4S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "5" or "E" format effector is recognized in output lines. No default.

CHANNEL_5_SEQUENCE (C5S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "4" or "D" format effector is recognized in output lines. No default.

CHANNEL_6_SEQUENCE (C6S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "3" or "C" format effector is recognized in output lines. No default.

CHANNEL_7_SEQUENCE (C7S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "9" or "I" format effector is recognized in output lines. No default.

CHANNEL_8_SEQUENCE (C8S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "X" or "J" format effector is recognized in output lines. No default.

CHANNEL_9_SEQUENCE (C9S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "Y" or "K" format effector is recognized in output lines. No default.

CHANNEL_10_SEQUENCE (C10S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "Z" or "L" format effector is recognized in output lines. No default.

CHANNEL_11_SEQUENCE (C11S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "W" or "M" format effector is recognized in output lines. No default.

CHANNEL_12_SEQUENCE (C12S)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "U" or "N" format effector is recognized in output lines. No default.

FORM_FEED_DELAY (FFD)

Supported by ASYNC and X.25 asynchronous TIPs only. Milliseconds (maximum 3000) the TIP must delay after sending a CHANNEL_x_ SEQUENCE value to the printer. Default is 1000 milliseconds. The TIP sends null characters to the device to effect the delay.

FOLD_LINE (FL)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Whether or not the TIP must fold lines longer than the device page width. Default is YES.

FORM_FEED_SEQUENCE (FFS)

Supported by ASYNC and X.25 asynchronous TIPs only. sequence of octets sent to the printer when a "1" or "A" format effector is recognized in output lines. Default is (0D(16),0C(16)) or (CR,FF).

KEYBOARD (K)

Supported by ASYNC and X.25 asynchronous TIPs only. Whether or not the printer has an associated keyboard. Default is NO.

If an asynchronous printer is configured with a keyboard, then you can use the keyboard to enter the following commands to control the printer: DROP, START,and STOP. See the CDCNET Batch Device User's Guide for more information onthese commands.

NO_SPACE_SEQUENCE (NSS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "+" format effector isrecognized in output lines. Default is 0D(16) (CR).

SINGLE_SPACE_DELAY (SSD)

Supported by ASYNC and X.25 asynchronous TIPs only. Milliseconds(maximum 1000) the TIP must delay after sending a SINGLE_SPACE_SEQUENCE to theprinter. Default is 50. The TIP sends null characters to the device to effect the delay.

SINGLE_SPACE_SEQUENCE (SSS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when one of the following format effectors is recognized in output lines:

Format Effector	Times Single Space Sequence Sent
" " (blank; single space)	One
"0" (double space)	Two
"–" (triple space)	Three

Default is (0D(16),0A(16)) or (CR, LF).

BOTTOM_OF_FORM_CHANNEL (BOFC)

Supported by ASYNC, HASP, and URI TIPs only. Channel to which the printer skips when a "2" or "B" format effector is recognized in output lines. May not be the same as the AUTO_PAGE_EJECT_CHANNEL (APEC) value. Default is 6.

VFU_TOP_FORM (VTF)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Whether or not the printer must be at top-of-form when the vertical format unit (VFU) load image is loaded. Default is YES.

MAXIMUM_VFU_LENGTH (MVL)

Supported by ASYNC, X.25 asynchronous, and URI TIPs only. Maximum lines in a VFU load image the printer supports. Default is 127 lines.

For a printer that supports VFU loading (that is, if the VFU_LOAD_OPTION parameter on DEFINE_BATCH_DEVICE is any value other than NONE), it is important that the MVL value does not exceed the maximum VFU length the printer actually supports. If the MVL value exceeds the actual supported length, attempts to load a VFU load image into the printer could fail. No files are sent to the printer until the problem is corrected.

INITIALIZATION_PROCEDURE (IP)

Name of the load procedure containing data output to the printer when the line activates. No default. Supported for any TIP handling transparent data.

INITIALIZATION_SEQUENCE (IS)

Sequence of octets sent to the printer when the line activates. If an initialization procedure is specified, the initialization sequence is sent after procedure data. No default. Supported for any TIP handling transparent data.

EIGHT_LPI_SEQUENCE (ELS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when a "T" format effector is recognized in output lines. No default. This parameter is allowed only if the DEFINE_BATCH_DEVICE VFU_LOAD_OPTION parameter value is NONE.

FILE_PREFIX_PROCEDURE (FPP)

Name of the load procedure containing data output to the printer before each nontransparent file. No default. Supported for any TIP handling transparent data.

NOTE

If you change an FPP, do the following to guarantee that all printers and printer terminal models using the FPP get a copy of the changed FPP:

For each printer terminal model that uses the FPP, send a CHANGE_PRINTER_MODEL_ATTRIBUTES command to the DI or DIs supporting the printers. Specify the TERMINAL_MODEL and FILE_PREFIX_PROCEDURE parameters. For more information, see the CDCNET Configuration Guide.

FILE_PREFIX_SEQUENCE (FPS)

Sequence of octets sent to the printer before each nontransparent file. If a file prefix procedure is specified, the file prefix sequence is sent after procedure data. No default. Supported for any TIP handling transparent data.

FILE_SUFFIX_SEQUENCE (FSS)

Sequence of octets sent to the printer after each nontransparent file. No default. Supported for any TIP handling transparent data.

SIX_LPI_SEQUENCE (SLS)

Supported by ASYNC and X.25 asynchronous TIPs only. Sequence of octets sent to the printer when an "S" format effector is recognized in output lines. No default. This parameter is allowed only if the DEFINE_BATCH_DEVICE VFU_LOAD_OPTION parameter value is NONE.

Responses Printer model <name> defined.

--ERROR-- Printer model <name> is already defined.

--ERROR-- The bottom of form and auto page eject channels cannot be the same.

--ERROR-- If EIGHT_LPI_SEQUENCE or SIX_LPI_SEQUENCE is specified, then both must be specified.

Remarks Use DEFINE_PRINTER_MODEL_ATTRIBUTES to specify printer attributes for a non-Control Data printer, or to create a new terminal model with different default printer attributes than the Control Data-supplied attributes.

NOTE

When defining new printer models, the terminal model name must be unique. Do not use any of the predefined, Control Data-supplied terminal model names, such as CDC_585V or CDC_CYBER18.

Use DEFINE_PRINTER_MODEL_ATTRIBUTES only for DIs connected to batch devices that use the terminal model defined by the command.

When defining a printer model for an asynchronous printer, you may have to define a file suffix sequence of (CF,LF) (using parameter FILE_SUFFIX_SEQUENCE=(CR,LF)), since some asynchronous printers require such codes to flush the printer buffer.

If an EIGHT_LPI_SEQUENCE or a SIX_LPI_SEQUENCE is defined for a printer terminal model, both must be specified. These sequences can only be used for batch devices having a DEFINE_BATCH_DEVICE VFU_LOAD_OPTION parameter value of NONE.

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 URI 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 do line folding. It supports a maximum VFU size of 255 lines.

```
define_printer_model_attributes terminal_model=non_585b ..
    bottom_of_form_channel=12 maximum_vfu_length=255
```

The next example defines an asynchronous printer without a VFU. It uses nonstandard preferences to control paper movement and initialize the printer.

```
define_printer_model_attributes ..
  terminal_model=myprinter ..
  file_prefix_sequence=(ESC,'R',CR) ..
  single_space_sequence=LF ..
  form_feed_sequence=ESC,'F' ..
  file_suffix_sequence=EOT
```

DEFINE_RECORDER_LOG_GROUP (DEFRLG) (NOS Only) DI Configuration Procedure/NETOU Command

Purpose

Defines the log recorder function in a NOS MDI or MTI. 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 Control Data 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 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 do not affect the Independent Log ME function for other trunks. Currently, only one recorder log group can be configured per channel trunk for each MDI/MTI system.

Format

DEFINE_RECORDER_LOG_GROUP
LOG_GROUP = name
PRIORITY = 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. Default 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 default, which is also the highest priority, 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. Default is the trunk specified on the DEFINE_SYSTEM command. Default on that command is the channel over which the MDI/MTI was loaded. If the MDI/MTI was not loaded over a channel, and no default channel trunk was specified on 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 is 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
```

DEFINE_REMOTE_LINE_MONITOR (DEFRLM)
DI Configuration/NETOU Command

Purpose Defines the title, forward-timer, LIM number, and port number of the Remote Line Monitor.

If the connection is broken, the Remote Line Monitor must be redefined.

Format

DEFINE_REMOTE_LINE_MONITOR
LINE_INTERFACE_MODULE = 0..7
PORT = 0..7
TITLE = name
FORWARD_TIMER = 200..30000

Parameters

LINE_INTERFACE_MODULE (LIM)

Defines the LIM number to be monitored.

PORT (P)

Defines the port number of the selected LIM to be monitored.

TITLE (T)

Defines the title that the Remote Line Monitor registers.

FORWARD_TIMER (FT)

Defines the timer to be used to forward monitor data in milliseconds. Default is 2000.

Responses

Remote Line Monitor defined, $LIM <lim>_PORT <port>.

Remote Line Monitor previously defined, $LIM <lim>_PORT <port>.

LIM <lim> and/or PORT <port> not available.

No line or trunk defined for $LIM <lim>_PORT <port>.

Unable to open Session Layer SAP.

Unable to register title, <title>.

Examples

```
define_remote_line_monitor lim=3 p=1 t=line1

Remote Line Monitor defined, $LIM5_PORT0.
```

DEFINE_REMOTE_LOAD_SUPPORT (DEFRLS) DI Configuration Procedure/NETOU Command

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.

If you have a network with several network solutions, you must establish a minimum number of DIs that provide remote load support, so that even DIs not directly connected to a NOS/VE or NOS host can be loaded. With remote load support, DIs can be loaded over a network solution to which they are directly connected.

NOTE

Do not use this command for an ICA-II.

This command is not required in DIs that are loaded over a NOS/VE or NOS host channel. For those DIs, remote load support is provided by default. This command is required to configure remote load support in DIs not directly connected to a NOS/VE or NOS host. For an example of remote load support, see the CDCNET Configuration Guide.

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. Default, which is also the highest priority, 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 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 does.

CONCURRENT_LOAD_LIMIT (CLL)

Specifies the maximum number of DIs which may be simultaneously loaded by the DI providing remote load support. Default 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 does 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 does not respond to load requests from DIs that are on the restricted network or networks. Default is an empty list; by default, a DI containing remote load support loads remote DIs over all directly connected network solutions.

Responses Remote Load Support is defined.

--ERROR-- Remote Load Support is already defined.

--ERROR-- Command 'define_remote_load_support' is invalid for an ICA.

--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

Remote Load Support is defined.
```

DEFINE_SERVER_DEVICE (DEFSD)

Purpose Defines both passthrough and device outcall servers. This command can be only executed after making a connection to the Outcall Gateway.

Format

DEFINE_SERVER_DEVICE
TITLE=name or *keyword*
ALLOWED_ACCESS=keyword
INACTIVITY_TIMER=120..14400 or *INFINITE*
CONTROL_SERVER_PORT=boolean
MODEM_PROTOCOL=name
PASSTHROUGH_RESPONSE=boolean

Parameters

TITLE (T)

Specifies the logical name of the server that must be used by clients when attempting to make connections to the server. Default is USER_ID.

When a name value is used and the server is site-configured, the value specified for this parameter is the title registered.

When a name value is used and a server is user-configured for passthrough access, the title registered is the value specified for this parameter with the prefix T_ (for example, T_name).

When a name value is used and the server is user-configured for device outcall access, the title registered is the value specified for this parameter with the prefix D_ (for example, D_name).

When a name value is used and a server is user-configured for both types of accesses, two titles are registered, one with each prefix.

When the key value USER_ID is specified, and the server port is configured for network validation, the title is constructed from the server's validated USER_NAME and DOMAIN_NAME. The title registered is in the form USER_NAME@DOMAIN_NAME.

When USER_ID is specified and the server port is not configured for network validation, the DEVICE_NAME associated with the server port is used as the title unless the DEVICE_NAME begins with a $ and the LINE_NAME does not. In this case, the LINE_NAME, rather than the DEVICE_NAME is used as the title.

ALLOWED_ACCESS (AA)

Specifies how the server can be accessed. The keyword value DEVICE_OUTCALL permits access from host applications. The keyword value PASSTHROUGH permits access from terminal users. The default keyword value ALL permits access to both, providing that the passthrough service has been purchased; if not, ALL is equivalent to the value DEVICE_OUTCALL.

INACTIVITY_TIMER (IT)

Specifies the maximum time, in seconds, that a client connection can remain idle. Allowed values are any integer from 120 through 14400, or the keyword INFINITE. Clients who do not transmit or receive data for this time period are disconnected from the server, and the server is made available to other clients. If you specify INFINITE, the network does not disconnect the connection because of inactivity.

The default value is determined by the DEFAULT_INACTIVITY_TIMER parameter of the DEFINE_OUTCALL_GATEWAY command.

CONTROL_SERVER_PORT (CSP)

Selects whether or not the control signals of the server port are disconnected when a client is not accessing the server. The default for user-configured servers is FALSE, and the default for site-configured servers is TRUE.

MODEM_PROTOCOL (MP)

Used only when the server device is a modem with dialout capability. This parameter specifies the command-level protocol implemented by the modem. Currently, the only supported keyword value is the keyword HAYES. This parameter has no default value.

When MODEM_PROTOCOL=HAYES, the outcall gateway requires that the client provide dialing information in the SERVICE_DATA parameter of the CREATE_CONNECTION command. The outcall gateway performs the dialing protocol with the modem and includes the dialed telephone number in the log messages that record server usage. You (the client) are limited to one call per connection and are not allowed to escape into command mode with the modem.

When the server device is a dialout modem and the MODEM_PROTOCOL parameter is not specified, you must perform the dialing protocol from your terminal. The outcall gateway discards any SERVICE_DATA that it receives, and does not record the dialed telephone numbers in the server usage logs.

PASSTHROUGH_RESPONSE (PR)

Specifies whether the server should pass input data immediately or wait for a <Carriage Return> or <Line Feed>. If TRUE, the server passes input data immediately, if FALSE, the server waits for a <Carriage Return> or <Line Feed> before passing input data. The default is FALSE.

NOTE

If configuring a postscript printer or other device that returns data, the PASSTHROUGH_RESPONSE parameter should be set to TRUE. If configuring an ASCII printer that does not return any data, the PASSTHROUGH_RESPONSE parameter can be left as the default of FALSE.

Examples

Following is an example command sequence for configuring a passthrough server:

```
create_connection $outcall
define_server_device allowed_access=passthrough
```

Since the TITLE parameter was not specified in this example, the default USER_ID is used, and the title registered depends on whether or not network validation is configured for this line. If it is, the title is constructed from the server's user name and domain name in the form USER_NAME@DOMAIN_NAME. If network validation is not configured, the title is the server's DEVICE_NAME, unless the DEVICE_NAME begins with a $. If it does, the server's LINE_NAME becomes the registered title.

NOTE

If the first input you send to the outcall gateway after the CREATE_CONNECTION command is not a DEFINE_SERVER_DEVICE command, you are prompted with the following message:

```
Input discarded.
Please enter DEFINE_SERVER_DEVICE command.
```

Responses

If the DEFINE_SERVER_DEVICE command is successful, the following response appears:

```
Server Device defined.
```

This response indicates that the user has become a registered passthrough server and, therefore, is a candidate to receive connections from passthrough clients.

When a passthrough client makes a connection to the server, the server receives the following message:

```
Passthrough client connection complete.
System = system_name, Line = line_name.
```

If the client subsequently disconnects from the server, the server receives the following message:

```
Client disconnected
```

If the site administrator stops the outcall gateway, the server and client receive the following message and their connections with the outcall gateway terminate:

```
Outcall Gateway stopped
```

The error responses to the DEFINE_SERVER_DEVICE command are:

```
Passthrough service is not available.
```

This indicates that the passthrough service feature has not been purchased (or installed).

```
Modem protocol is not supported.
```

This message indicates that the value specified for the MODEM_PROTOCOL parameter is not HAYES.

If input is sent by the server when no client is connected, the server receives the following message:

```
Input discarded
Waiting for Client
```

DEFINE_SERVER_TELNET_GW (DEFSTG)
DI Configuration Procedure/NETOU Command

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) or the TCP/IP access interface 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

The host service may impose inactivity limits.

Format

DEFINE_SERVER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255
TITLE = name
TRANSLATION_DOMAIN = name
MAX_CONNECTIONS = 0..65535 or *INFINITE*
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..7FFFFFFF(16)
TCP_TIMEOUT = 0..65535 or *INFINITE*
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 TCP/IP, 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 (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 TITLE parameter of the DEFINE_NP_TERMINAL_GW command. If the destination system is NOS/VE, this title must be the one registered by the terminal manager. Default is supplied from the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the CDCNET catenet that should be searched for the service corresponding to the title information given in the TITLE parameter. Default 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. Default is INFINITE.

TCP_PORT_NUMBER (TPN)

Specifies the TCP port number to be used by the gateway. 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 can increase the number of instances of DI congestion. Default is 4096 bytes.

CAUTION

Changing this value is discouraged because network service might be disrupted.

TCP_TIMEOUT (TT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgement of data transmission. If an acknowledgement is not received within the specified period, TCP aborts the connection. A small TCP timeout value (less than a few seconds) might cause frequent and unnecessary loss of service during periods of network congestion. A large TCP timeout value might leave users waiting a long period of time after a host or network has failed. If INFINITE is entered, the connection does not time out. Default is 300 seconds.

START (S)

Specifies that the newly configured gateway is to be started after it is defined. Default 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_SERVICE (DEFS)
INETD Configuration File Command

Purpose Creates an entry in the INETD_CONFIGURATION file for a service that Internet Daemon (INETD) controls.

This command is valid only in file $SYSTEM.TCP_IP.INETD_CONFIGURATION. Do not use this command in a DI configuration procedure.

Format

DEFINE_SERVICE
  SERVICE = name
  ALIASES = list of names
  APPLICATION_NAME = name
  DEBUG_MODE = boolean
  DEBUG_MODE_INTERACTIVE = boolean
  INITIAL_STATE = keyword value
  MANTA_FILE_NAME = file
  PARAMETER_STRING = string
  PORT = integer
  PROTOCOL = keyword value
  QUEUE_LIMIT = integer 1..15
  SERVICE_TASK = file
  STARTING_PROCEDURE = name
  USER_FLAG_1 = boolean
  USER_FLAG_2 = boolean
  USER_FLAG_3 = boolean
  WAIT = boolean

Parameters

SERVICE (S)

Service name.

ALIASES (A)

List of aliases which may be used to access the service.

APPLICATION_NAME (AN)

Application name as it is known to IPAM and NAM/VE; used by NAM/VE for connections.

DEBUG_MODE (DM)

If TRUE, defines debug mode of the service task options passed to the task. Default is FALSE.

DEBUG_MODE_INTERACTIVE (DMI)

Causes the system task to be initiated under the NOS/VE system debugger. Default is FALSE.

INITIAL_STATE (IS)

Specifies if the server is initialized as an active or inactive service. Keyword values are ACTIVE and INACTIVE. Default is INACTIVE.

MANTA_FILE_NAME (MFN)

Name of the file used for communication between MANTA and the server. File is created by the service during initialization.

PARAMETER_STRING (PS)

Character string of data that can be passed to the server task by INETD. For more information about passing data to the REXEC server task, see the default job class information in the NOS/VE Network Management manual. For more information about passing data to the SMTP, TFTP, and LPD server tasks, see the TCP/IP Internet Daemon (INETD) information in the CDCNET Configuration manual.

PORT (P)

Well-known port number at which INETD listens for incoming connection requests or UDP packets.

PROTOCOL (P)

Specifies the protocol to be used for the service. Allowed keyword values are TCP and UDP. Default is UDP.

QUEUE_LIMIT (QL)

Valid only with TCP protocol. Specifies the queue limit on the call to IPAM. Default is 5. For further information, see the CDCNET TCP/IP Programming Interfaces and Applications manual.

SERVICE_TASK (ST)

Service binary file which is executed as a system task.

STARTING_PROCEDURE (SP)

Name of the procedure within the server to which control is passed after load completion.

USER_FLAG_1 (UF1)

These flags can be set in the service task options which are passed to the server task. The purpose of these flags is server-independent. Default for all three is FALSE.

USER_FLAG_2 (UF2)

See USER_FLAG_1.

USER_FLAG_3 (UF3)

See USER_FLAG_1.

WAIT (W)

Valid only for UDP ports. Set to TRUE if the service is defined as single-threaded and handles all datagrams arriving on the port. Set to FALSE if the service is defined to be multi-threaded, which frees the port for further use.

Examples

```
"INETD configuration file entry for the FTP server."

  Define_service ..
    service=ftp ..
    application_name=OSA$FTP_SERVER ..
    debug_mode=off ..
    debug_mode_interactive=off ..
    initial_state=active ..
    port=21 ..
    protocol=TCP ..
    service_task=$SYSTEM.TCP_IP.FTP_BOUND_PRODUCT ..
    starting_procedure=IPP$FTP_SERVER_TASK ..
    manta_file_name=$SYSTEM.TCP_IP.FTP_MANTA_SERVICE_FILE

"INETD configuration file entry for the SMTP server."

  Define_service ..
    service=smtp ..
    application_name=OSA$SMTP_SERVER ..
    debug_mode=off ..
    debug_mode_interactive=off ..
    initial_state=active ..
    parameter_string='MDU=MVE' ..
    port=25 ..
    protocol=TCP ..
    service_task=$SYSTEM.TCP_IP.SMTP_BOUND_PRODUCT ..
    starting_procedure=IPP$CREATE_SMTP_RECV_CONNECTION ..
    manta_file_name=$SYSTEM.TCP_IP.SMTP_MANTA_SERVICE_FILE
```

DEFINE_SLIP_GW (DEFSG)
DI Configuration Procedure/NETOU Command

Purpose

Configures a Serial Line Internet Protocol (SLIP) gateway. The SLIP gateway interfaces to the IP router in the DI. This provides the workstation with access to all IP networks known to the DI.

If the IP address does not specify an existing IP network in this DI, a new network is created. You must issue a DEFINE_IP_HOST command to define the local IP address for the new network.

To access the SLIP gateway, issue a CREATE_CONNECTION terminal user command. See the CDCNET Terminal Interface manual for more information.

Format

DEFINE_SLIP_GW
GATEWAY_NAME = name
IP_NETWORK = list 1..4 of 0..255
MAX_DATAGRAM_SIZE = 20..1500
SUBNET_MASK_SIZE = 10..30
TITLE = list 1..15 of name
MAX_SLIP_HOSTS = 1..200
INACTIVITY_TIMEOUT = 0..65535 or INFINITE
LOCAL_LINES_ONLY = boolean
START = boolean

Parameters

GATEWAY_NAME (GN)

Specifies the logical name of the network SLIP gateway used by subsequent commands that reference the gateway. This parameter is required.

IP_NETWORK (IN)

Specifies the network/subnet number assigned to the SLIP gateway in the DI which may or may not specify an existing network. If a new network/subnet is specified, the network is defined. If an existing network/subnet is specified, connected SLIP host addresses are available to other systems via ARP requests. This parameter is required.

MAX_DATAGRAM_SIZE (MDS)

Specifies the maximum datagram size in bytes that the SLIP network can handle without fragmentation. If the IP_NETWORK parameter specifies an existing network, this parameter is ignored. The default is 1500 bytes.

SUBNET_MASK_SIZE (SMS)

Specifies whether the SLIP network is to be subnetted. The value represents the number of bits in the subnet mask. For subnetting to occur, the subnet mask must be at least 10, 18, or 26 for Class A, B, or C networks, respectively. The subnet mask size must be the same for all subnets of a given IP network. If the IP network has been defined to be subnetted (with a previously issued DEFINE_IP_NET or DEFINE_SLIP_GW command) and this parameter is omitted, the previously defined subnet mask size is used. Otherwise, the IP network defined is not subnetted.

TITLE (TITLES) (T)

Specifies the title(s) by which this gateway service can be accessed. For example, this is the name the CDCNET terminal users supply in the CREATE_CONNECTION command. The default is the value supplied for the GATEWAY_NAME parameter.

MAX_SLIP_HOSTS (MSH)

Specifies the maximum number of simultaneous SLIP connections to be supported by the gateway. The default is 100.

INACTIVITY_TIMEOUT (IT)

Specifies the interval, in seconds, a connection can be idle before the network connection is terminated. The default is INFINITE.

LOCAL_LINES_ONLY (LLO)

Specifies whether the CREATE_CONNECTION must come from a line connected to the DI (that is, the SLIP gateway and the line using it are in the same DI). The default is FALSE.

START (S)

Specifies whether the newly configured gateway is to be started after it is defined. The default is TRUE.

Responses

SLIP gateway <gateway_name> is defined and started for <new OR existing> <network OR subnet> <network_address>.

SLIP gateway <gateway_name> is defined for <new OR existing> <network OR subnet> <network_address>.

--ERROR-- SLIP gateway <gateway_name> is already defined.

--ERROR-- IP subnet <ip_network> specified is too large for subnet mask size.

--ERROR-- IP subnet <ip_network> is invalid.

--ERROR-- IP network <ip_network> is not directly connected.

--ERROR-- IP network <ip_network> subnet mask size is inconsistent with the previously defined subnet mask size.

--ERROR-- SLIP gateway title <title> is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_slip_gw gateway_name=slip_gw ip_network=(129,179,71) ..
  subnet_mask_size=24

SLIP gateway SLIP_GW id defined and started for a new subnet
129.179.071.000.

define_slip_gw gateway_name=sl_gw_2 ip_network=(129,179,72,1) ..
  subnet_mask_size=24 start=false

SLIP gateway SL_GW_2 is defined for an existing subnet 129.179.072.000.
```

```
define_slip_gw gateway_name=slip_gw ip_network=(129,179,71,1) ..
  subnet_mask_size=24

--ERROR-- SLIP gateway SLIP_GW is already defined.
```

DEFINE_SLIP_HOST (DEFSH)
DI Configuration Procedure/NETOU Command

Purpose Adds an entry to the specified SLIP gateway which relates the specified host name to an IP address or range of IP addresses. If the host count is greater than 1, the IP addresses assigned are the consecutive values starting with the specified IP_ADDRESS parameter. Use the CREATE_CONNECTION terminal user command to access the SLIP gateway. See the CDCNET Terminal Interface manual for more information. Include a SLIP host name as the SERVICE_DATA parameter to assign the workstation the corresponding IP address for the SLIP connection.

Format

DEFINE_SLIP_HOST
HOST_NAME = name
GATEWAY_NAME = name
IP_ADDRESS = list 1..4 of 0..255
HOST_COUNT = 1..100
HOST_NAME_STRING = string 0..255

Parameters

HOST_NAME (HN)

Specifies the logical name to be assigned to this entry. This parameter is required.

GATEWAY_NAME (GN)

Specifies the logical name of the SLIP gateway for which this entry is used. This parameter is required.

IP_ADDRESS (IA)

Specifies the IP host address to be assigned to the user side of the SLIP network. The network and subnet portions of the IP address must be the same as the IP address from the corresponding DEFINE_SLIP_GW command. This parameter is required.

HOST_COUNT (HC)

Specifies the number of hosts for this entry. IP addresses are assigned in consecutive order starting with the specified IP_ADDRESS value. The default is 1.

HOST_NAME_STRING (HNS)

Specifies the string to be included in the text returned to the workstation when a SLIP connection is set up. This occurs after a CREATE_CONNECTION is made to the SLIP gateway from a workstation. The first character of the string is treated as a format effector to allow control of the vertical spacing for the string. In the default value shown below, *host_adr* is the IP address assigned to the SLIP host and the *gw_adr* is the local IP address of the SLIP network in the DI

```
Attaching <host_name> (<host_adr>) to domain CDCNET via <gateway_name> (<gw_adr>)
```

NOTE

This message preceded by SUNSLIP is required to cause SLIP in PC/NFS dial-up mode to become active.

Responses

SLIP host <host_name> added to gateway <gateway_name>.

--ERROR-- SLIP host <host_name> is already defined for SLIP gateway <gateway_name>.

--ERROR-- SLIP gateway <gateway_name> is not defined.

--ERROR-- IP address <ip_address> is invalid.

--ERROR-- IP address <ip_address> is already defined.

--ERROR-- SLIP host address <ip_address> does not correspond to the network address <ip_address> for SLIP gateway <gateway_name>.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_slip_host gateway_name=slip_gw host_name=host_2 ..
  ip_address=(129,179,71,2)

SLIP host HOST_2 added to gateway SLIP_GW.

define_slip_host gateway_name=slip_gw host_name=host_3 ..
  ip_address=(129,179,71,3) host_count=10

SLIP host HOST_3 added to gateway SLIP_GW.

define_slip_host gateway_name=slip_gw host_name=host_2 ..
  ip_address=(129,179,71,2)

--ERROR-- SLIP host HOST_2 is already defined for SLIP gateway SLIP_GW.

define_slip_host gateway_name=slip_xx host_name=host_2 ..
  ip_address=(129,179,71,2)

--ERROR-- SLIP gateway SLIP_XX is not defined.

define_slip_host gateway_name=slip_gw host_name=host_22 ..
  ip_address=(129,179,71,2)

--ERROR-- IP address 129.179.71.2 is already defined.

define_slip_host gateway_name=slip_gw host_name=host_22 ..
  ip_address=(129,179,71,246) host_count=20

--ERROR-- IP address 129.179.72.9 is invalid.

define_slip_host gateway_name=slip_gw host_name=host_2 ..
  ip_address=(129,179,77,2)

--ERROR-- SLIP host address 129.179.77.2 does not correspond
to the network address 129.179.71 for SLIP gateway SLIP_GW.
```

DEFINE_SNMP_AGENT (DEFSA)
DI Configuration Procedure/NETOU Command

Purpose

Configures the SNMP agent in a CDCNET communications device. The DI SNMP agent provides CDCNET users who have TCP/IP networks the ability to monitor, control and manage their CDCNET devices by any RFC-compliant SNMP Manager.

This command defines a *public* community as any that accepts PDUs from IP addresses with read-only access modes. Only CDCNET allows commands of length less than 256. If necessary, you can use CHANGE_SNMP_AGENT to set any variables that do not fit on the DEFINE_SNMP_AGENT command. These values can also be changed by authenticated SNMP Managers with read-write access. The variables defined in MIB II (RFC 1213) and the CDCNET enterprise-specific MIB are supported. The SNMP Agent uses UDP to send and receive SNMP PDUs.

Format

DEFINE_SNMP_AGENT
NAME = string 1..226
CONTACT = string 1..226
LOCATION = string 1..226
MAX_PDU_SIZE = integer 484..8192

Parameters

NAME (N)

Specifies the administratively assigned name for this communications device. This parameter is optional. There is no default.

CONTACT (C)

Specifies the contact person for this communication device, together with information on how to contact this person. This parameter is optional. There is no default.

LOCATION (L)

Specifies the physical location of this communication device. This parameter is optional. There is no default.

MAX_PDU_SIZE (MPS)

Specifies the maximum size PDU that an SNMP Agent can build and send to the peer SNMP Manager. The default is 8192, the maximum size support by the underlying UDP protocol.

Responses

The SNMP Agent has been defined.

--ERROR-- The UDP Interface is not defined.

--FATAL-- Not enough memory is currently available for required table space.

--WARNING-- The SNMP Agent is already defined.

Remarks

Programmers can find a listing of the CDCNET MIB II in file $SYSTEM.CDCNET.VERSION_xxxx.SNMP.CDCNET_MIB (where xxxx is the current CDCNET version). This file contains a listing of the VAR and CONST used in the CDCNET MIB II. Sites can use the VAR and CONST as necessary for SNMP operations.

DEFINE_SOURCE_ALARM_MESSAGE (DEFSAM)
DI Configuration Procedure/NETOU Command

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 are 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 omitted, a set of default alarm message numbers are enabled. See the online list of default log and alarm messages for the alarm message numbers and their message identifiers. The CDCNET Configuration Guide further explains how to access these messages. You may add alarms to this list using additional DEFSAM commands. You may also cancel messages using the CANCEL_SOURCE_ALARM_MESSAGE command.

NOTE

Canceling any of the default alarms is not recommended.

For the complete list of diagnostic messages, see the online 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).

Examples

```
define_source_alarm_message

Source alarm messages defined.
```

This page was intentionally left blank.

DEFINE_SOURCE_LOG_GROUP (DEFSLG)
DI Configuration Procedure/NETOU Command

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 are logged by the DI.

Format

DEFINE_SOURCE_LOG_GROUP
LOG_GROUP = 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. Default 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. The CDCNET Configuration Guide explains how to access these messages. You may add or delete log message numbers to this list using the CHANGE_SOURCE_LOG_GROUP command.

NOTE

Omitting any of the default set of messages is not recommended.

For the list of diagnostic messages and their numbers, see the online 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)
DI Configuration Procedure

Purpose Defines a 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..8000(16)
STANDARD_STACK_SIZE = 0800(16)..2000(16)
DEFAULT_CHANNEL_TRUNK = name
ROUTING_SYSTEM = boolean
CLOCKING_SYSTEM = boolean
MAXIMUM_RECOVERIES = 0..255
COMPRESS_DUMP = boolean

Parameters

SYSTEM_NAME (SN)

Title of a DI or an Integrated Communications Adapter (ICA) board, as the title appears in the CDCNET directory. For a DI, the default system 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. For an ICA-II board, the default system name is $ICA_*sysid*, where *sysid* is the 12-hexadecimal-digit unique system address.

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. 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.

The actual buffer size saved 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

Default is 144 for a DI with less than 4 megabytes of SMM and 524 for an ICA-II with 4 or more megabytes of SMM (set at power-up but can be altered if this parameter is specified).

BUFFER_PERCENTAGE (BP)

Sets the percentage of total System Main Memory (SMM) which, after DI configuration, is to be turned initially into data buffers. Default is 50 percent.

The free memory remaining after the data buffers are allocated is called allocatable memory. A DI uses allocatable memory to configure and activate devices and to load transient processes including command processors. If a DI is short of allocatable memory, device configuration may fail or processes may fail to load.

You may increase the amount of allocatable memory for a DI by decreasing its buffer percentage. If all but a few of the devices defined for a DI are successfully configured and activated, enough allocatable memory to activate all devices may be found by decreasing the buffer percentage. Slowly decrease the buffer percentage until all devices successfully are configured. A DI needs at least 100 K bytes of total system buffers (700 buffers at default buffer size of 144 bytes) to ensure that there are enough buffers for normal system operation.

Control Data also recommends that the buffer percentage be at least 30 percent. Below 30 percent, a DI may not have enough buffers to service with good performance all the configured devices. A shortage of buffers may cause long delays in data transmission or poor response for interactive terminal users.

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. Control Data recommends that you keep this at 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 2. 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 8. 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)

Identifies a system to be a distributor of Routing Information Data Units (RIDUs) when the value is set to TRUE. Systems with ROUTING_SYSTEM set to TRUE generate and forward all the RIDUs and version 3 directory PDUs. See the CHANGE_DIRECTORY command for information on specifying the Directory ME version.

If the value is FALSE, the system does not generate or forward RIDUs or version 3 directory PDUs from other systems. If the primary (lowest cost) network is inactive, the system only generates RIDUs, it does not forward them. Setting ROUTING_SYSTEM to FALSE for all but one or two DIs on the redundant Ethernets reduces the amount of unnecessary RIDU traffic.

Default is TRUE.

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. Default 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.

NOTE

Do not use this parameter for an ICA-II.

MAXIMUM_RECOVERIES (MR)

Specifies the maximum number of task recoveries allowed in a 60-minute period before the DI is reset. Default is 5.

COMPRESS_DUMP (CD)

Requests a compressed dump (omits free memory and buffers) of DI memory. Possible parameter values are YES and NO. Default is NO.

Responses

The DEFINE_SYSTEM command can only be executed in a system configuration procedure. The following response 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-- Clocking_system may not be specified for an ICA II system.

--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

- You should allow for at least 100 K of memory (about 700 buffers) for system buffers to be available after system configuration.

  CAUTION

  Use only default values for parameters BUFFER_BOUNDARY_PERCENTAGE through STANDARD_STACK_SIZE. Changing these values may significantly degrade performance.

- This command is not required to be present in a DI's system configuration file. If this command is not present, default values are internally generated during initialization.
- To change values of a DI's operating system while the DI is operational, use the CHANGE_SYSTEM network operator command.

Examples

```
define_system system_name=mdi_86

The define system command is completed.
```

DEFINE_TCP_INTERFACE (DEFTI)
DI Configuration Procedure/NETOU Command

Purpose Configures the TCP interface (DOD's Transmission Control Protocol). This command is required if a DI supports TCP/IP.

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
SYN_ACK_TIME = 2..10000
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.
IN_WINDOW (IW)	Segments are accepted if they fall within the current TCP window. The TCP window is the amount of data a receiving TCP entity is willing to accept from the sending TCP entity in bytes. Segments outside the TCP window are discarded.

Default is IN_WINDOW.

CAUTION

Using this parameter might cause performance degradation and increase the number of retransmitted segments.

ACK_PERCENTAGE (AP)

Specifies the percentage of the receive window that must be full before an acknowledgement is issued. Default is 50.

MAX_BUFFERS (MB)

Specifies the maximum number of data bytes that TCP holds for a connection for both directions of travel. Default is 16,384 bytes.

MAX_SEGMENT_SIZE (MSS)

Specifies the maximum segment size, in bytes, to be negotiated by TCP when a connection is established. If the maximum datagram size of the first hop IP network is less than this value, the maximum datagram size is used as the maximum segment size. Otherwise, the value for this parameter is used. Default is 1456 bytes.

MAX_CONNECTIONS (MC)

Specifies the maximum number of simultaneous TCP connections. This includes active and passive connections. Default 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 identifier. 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 does not respond or acknowledge connection establishment. Default 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 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 acknowledgement before retransmitting a data segment. This value changes for an active connection as the actual round-trip time is learned. Default 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.
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.

CAUTION

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. Default is 60 hops.

SYN_ACK_TIME (SAT)

Specifies the maximum number of seconds that TCP should wait for an acknowledgement of the initial SYN segment. The connection is aborted at timeout. A SAT time too small can cause connection disconnects before the remote host has time to respond. A large SAT time can cause users to wait long periods of time for an unavailable host. Default is 60 seconds.

START (S)

Specifies that the TCP task should be started and connection attempts are honored. Default 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) (NOS Only) DI Configuration Procedure/NETOU Command

Purpose Configures a gateway that provides services to NOS CYBER-resident TCP/IP applications, such as File Transfer Protocol (FTP). Gateways for the TCP, IP, UDP, and TELNET protocols can be configured. This command is not needed to support TELNET interactive services.

NOTE

For NOS/VE applications, use the ADDTA command.

Format

DEFINE_TCPIP_GW
GATEWAY_NAME = name
SOURCE_IP_ADDRESS = list 4 of 0..255
TITLE = list 1..15 of name(s)
TRANSLATION_DOMAIN = name
PROTOCOL = list 1..4 of keyword value
MAX_CONNECTIONS = 0..65535 or *INFINITE*
START = boolean

Parameters

GATEWAY_NAME (GN)

The logical name of the gateway used in subsequent commands that reference the gateway.

SOURCE_IP_ADDRESS (SIA)

The IP address of the CYBER host for which this gateway provides service. This parameter is the source IP address used when the host application does not specify one. If the host application specifies a source IP address and this parameter is specified, the host source IP address is used. If the host application does not specify a source IP address and this parameter is omitted, the IP address defined for the DI is used.

The format for this parameter is similar to the decimal octet convention used by TCP/IP, 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 (T)

Specifies the title that the host applications must use to access this gateway. This title must be coordinated with the Internet Protocol Access Method (IPAM) installation on NOS/VE, or the configuration of the NDL OUTCALL on NOS. Default is the value supplied for the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the catenet for which gateway services are to be made available. Default is CATENET.

PROTOCOL (P)

Specifies the protocol supported by this gateway. The allowed keyword values are TCP and UDP which provide host interfaces for the respective protocols. This parameter controls which piece of the gateway is loaded. FTP requires the TCP interface. Default is TCP.

NOTE

User written applications may need TCP, UDP, or IP.

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. Default is 65535.

START (S)

Specifies that this gateway is to be started. Default 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_106 ..
protocol = tcp

TCP/IP gateway FTP_GW is defined and started to support TCP protocol.
```

DEFINE_TIP (DEFT)
DI Configuration Procedure/NETOU Command

Purpose Defines the following for each TIP:

- A CDCNET TIP name or an optional user-defined TIP name that sites use to redefine TIP names.
- 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.

You can specify certain parameters based on a TIP, line, or terminal device. These overlapping parameter definitions allow you to set values on the TIP rather than specifying individual parameter definitions for each line that the TIP is to support. Identical parameters on the DEFINE_LINE, DEFINE_TERMINAL_DEVICE, and ADD_TELNET_SERVER commands override the parameter settings on DEFINE_TIP.

To define the X.25 Asynchronous TIP, use the DEFINE_X25_ASYNCTIP command instead of DEFINE_TIP.

To define the SNA3270 TIP, use the DEFINE_SNA3270_TIP 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
VALIDATE_USERS = boolean

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
NTFTIP or NTF
USER1TIP or USER1
USER2TIP or USER2
USER3TIP or USER3
USER4TIP or USER4
MODE4TIP or MODE4
NJEFTIP or NJEF
TELNETTIP or TELNET

If a TIP is not loaded as part of the DI load process, the first terminal user is likely to encounter a long delay (up to a minute on a heavily loaded system) before the terminal is recognized. This delay is due to the dynamic loading of the TIP. To avoid the delay, include a LOAD_MODULE command for each TIP configured in a DI in the DI's system configuration procedure. The following is an example LOAD_MODULE command for the ASYNCTIP:

```
LOAD_MODULE ASYNCTIP_MODULE
```

USER_TIP_NAME (UTN)

This parameter is used for sites that implement user TIPs (TIPs developed at the site rather than provided with Control Data release software) and want 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 specify one of the user TIPs. When the USER_TIP_NAME parameter is specified along with one of the user TIP names, 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. The value for this parameter can be no more than 26 characters long.

LINE_CONTROL_SUPPORT (LCS)

Specifies the level of line control required by the TIP of the Line Control Module (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 LCM to perform the CIM configuration of the line, but that the TIP monitors 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.

Default is FULL.

NOTE

For TIPs currently provided with CDCNET using the DEFT command, 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	ASYNC
HASPTIP, NJEFTIP, NTFTIP, BSC3270TIP, MODE4TIP	SYNC
URITIP	PARALLEL
Non-Control Data provided TIPs	ASYNC

CLUSTER_ADDRESS (CA)

This parameter is used for BSC3270 and MODE4 TIPs. Specifies the cluster address to be used by the TIP for communication with an implicitly defined console device, that is, a console is implicitly defined when a line becomes active and no TDP has been defined by the DEFL command. This parameter does NOT specify a default value for subsequent DEFTD or DEFBD commands defined in a TDP for BSC3270 or MODE4 supported devices. Default is 0.

For the MODE4 TIP, the CLUSTER_ADDRESS must be in the range of 70(16) through 7F(16) for Mode 4A and 20(16) through 7F(16) for Mode 4C clusters. This parameter must be specified for the MODE4 TIP if any Mode 4 lines are defined by a DEFL without a TDP that defines the first console device.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the cluster address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

DEVICE_ADDRESS (DA)

This parameter is used for HASP, BSC3270 and MODE4 TIPs. Specifies the device address to be used by the TIP for communication with an implicitly defined console device, that is, a console is implicitly defined when a line becomes active and no TDP has been defined by the DEFL command. This parameter does NOT specify a default value for subsequent DEFTD or DEFBD commands defined in a TDP for BSC3270 or MODE4 supported devices. Default is 0.

For devices supported by the HASP TIP, the DEVICE_ADDRESS parameter is ignored if the 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. The default is 1 for HASP batch devices.

For the MODE4 TIP, the DEVICE_ADDRESS must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. This parameter must be specified for the MODE4 TIP if any Mode 4 lines are defined by a DEFL without a TDP that defines the first console device.

For the BSC3270 TIP, the device address must be in the range of 0 through 31. Default is 0.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the cluster address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

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. Default for this parameter varies according to the TIP being defined, as shown in the following list.

TIP Name	Default Transmission Block Size
HASPTIP, NTFTIP, BSC3270TIP	400
ASYNCTIP	450
NJEFTIP	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).

This value reserves input buffer space in the CIM. Use the maximum block size sent to CDCNET by a HASP, Mode 4, 3270 BSC, NJF, or NTF connection. Otherwise, CDCNET may disconnect the line because of excessive errors on the line.

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 lets you 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. The value of this parameter on this command can 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. There is no default.

VALIDATE_USERS (VU)

Specifies if users of this TIP are validated when validation is enabled by the DEFINE_USER_VALIDATION command. If TRUE is specified, validation is enabled. If FALSE is specified, the user of this TIP is not validated. Default is TRUE.

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

TIP HASPTIP defined.
```

DEFINE_UDP_INTERFACE (DEFUI)
DI Configuration Procedure/NETOU Command

Purpose Configures the User Datagram Protocol (UDP) layer interface. This command is required if the DI is to support TCP/IP applications that use UDP as the underlying transport service.

NOTE

This command is required to configure UDP to support the Network File System (NFS).

Format

DEFINE_UDP_INTERFACE
SECURITY_CHECKING = keyword value
TIME_TO_LIVE = 0..255
START = boolean

Parameters *SECURITY_CHECKING (SC)*

Specifies the security checking to be performed on all datagrams. 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 a passive or active connect request establishes the security level of the connection.
LEVEL_U (LU)	All connections must be at security level of UNCLASSIFIED.
LEVEL_C (LC)	All connections must be at security level of CONFIDENTIAL.
LEVEL_E (LE)	All connections must be at security level of EFTO.
LEVEL_M (LM)	All connections must be at security level of MMMM.
LEVEL_P (LP)	All connections must be at security level of PROG.
LEVEL_R (LR)	All connections must be at security level of RESTRICTED.
LEVEL_S (LS)	All connections must be at security level of SECRET.
LEVEL_T (LT)	All connections must be at security level of TOP SECRET.

Default is NONE. If a security level is specified, all incoming datagrams which have the security option supplied must match this security level. Any datagrams that do not match the security level are discarded.

TIME_TO_LIVE (TTL)

Specifies the IP time-to-live field used by UDP. In the TCP/IP community, this is typically used as a hop count. When the count in a packet reaches zero, the packet is discarded to prevent looping. Default is 60.

START (S)

Specifies that the UDP should be started and service requests is honored. The default value is TRUE.

Responses

UDP interface is defined and started.

UDP interface is defined.

--ERROR-- UDP interface is already defined.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_udp_interface . .
   time_to_live = 30
```

UDP interface is defined and started.

DEFINE_USER_TELNET_GW (DEFUTG)
DI Configuration Procedure/NETOU Command

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 can configure a user TELNET gateway with the IP address of the remote host which is accessed via that gateway. A user TELNET gateway can also be defined with the IP address omitted or specified as NONE. The terminal user, in this case, must supply the IP address of a remote host when creating a connection to that user TELNET gateway. For a description of the CREATE_CONNECTION (CREC) command, including using the SERVICE_DATA parameter to specify the IP address, see the CDCNET Terminal Interface manual.

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

The host service might impose inactivity limits of its own.

Format

DEFINE_USER_TELNET_GW
GATEWAY_NAME = name
IP_ADDRESS = list 4 of 0..255 or *keyword*
TITLE = list 1..15 of name
TRANSLATION_DOMAIN = CATENET
MAX_CONNECTIONS = 0..65535 or *INFINITE*
SOURCE_IP_ADDRESS = list 4 of 0..255
TCP_PORT_NUMBER = 0..65535
TCP_ALLOCATE_SIZE = 0..7FFFFFFF(16)
TCP_TIMEOUT = 0..65535 or *INFINITE*
INACTIVITY_TIMEOUT = 0..65535
INITIATE_OPTION_NEGOTIATION = boolean
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 that provides the TELNET interactive service. This user TELNET gateway establishes a connection using this IP address as the destination address. If this parameter is omitted or the keyword NONE is specified, the terminal user must specify the IP address of the remote host as "service data" when creating a connection (CREATE_CONNECTION) to the user TELNET gateway. The format of the IP address 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. Default is the value supplied for the GATEWAY_NAME parameter.

TRANSLATION_DOMAIN (TD)

This parameter specification is allowed but the value is ignored. Specifies the portion of the CDCNET catenet that can access this service. Default 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. Default 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). Default 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.

CAUTION

Specifying this value is discouraged, because poor network service might result.

Default 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.

CAUTION

A small TCP timeout value (less than a few seconds) might cause frequent and unnecessary loss of service during periods of network congestion. A large TCP timeout value might leave users waiting a long period of time after a host or network has failed.

If INFINITE is entered, the connection does not time out. Default 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. Default is 600 seconds.

INITIATE_OPTION_NEGOTIATION (ION)

Allows CDCNET to inhibit the negotiation of Telnet options on the TCP connections made by the User_Telnet_Gateway. Default is TRUE.

TRUE
CDCNET initiates the negotiation of Telnet options on the TCP connections made by the User_Telnet_Gateway.

FALSE
CDCNET inhibits the negotiation of Telnet options on the TCP connections made by the User_Telnet_Gatway.

START (S)

Specifies that the newly configured gateway is to be started after it is defined. Default is TRUE.

Responses

User TELNET gateway <gateway_name> is defined and started.

User TELNET gateway <gateway_name> is 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 _USER _VALIDATION (DEFUV)
DI Configuration Procedure/NETOU Command

Purpose Designates validation for interactive users connecting to CDCNET via communications lines, X.25 trunks, or Telnet connections. It also defines the default validation domain for the DI and the number of login attempts allowed before disconnecting a user.

Format

DEFINE _USER _VALIDATION
DEFAULT _VALIDATION _DOMAIN = name
MAXIMUM _LOGIN _ATTEMPTS = 1..10

Parameters

DEFAULT _VALIDATION _DOMAIN (DVD)

Specifies the default validation domain that users connected to this DI belong to by default.

You can override this default by specifying your own default validation domain during login.

MAXIMUM _LOGIN _ATTEMPTS (MLA)

Specifies the number of consecutive, unsuccessful login attempts allowed before disconnecting a user from the network. Default is three attempts.

Responses User validation is defined.

Examples

```
define_user_validation default_validation_domain=valexp1

User validation is defined.
```

DEFINE_VE_INTERFACE (DEFVI)
DI Configuration Procedure/NETOU Command

Purpose Defines the VE interface to a NOS/VE host.

NOTE

An "unable to start" error leaves the interface defined but not started.

Format

DEFINE_VE_INTERFACE
INTERFACE_SLOT = 0..7
INTERFACE_NAME = name
MAXIMUM_FRAME_SIZE = 4424..13000
NETWORK_ID = 1..0FFFF(16)
CHANNEL_TIMEOUT = boolean
START = boolean

Parameters

INTERFACE_SLOT (IS)

The number of the physical board slot which houses the MCI board. This parameter does not apply to an ICA-II. This parameter is **required** if more than one MCI board exists in the DI and is *optional* if only one MCI board exists in the DI.

NOTE

Do not use this parameter for an ICA-II.

INTERFACE_NAME (IN)

The logical name of the interface. The default name for an MDI is constructed using the MCI slot parameter, as in $MCI2. The default name for an ICA-II is $ICA_ICI.

MAXIMUM_FRAME_SIZE (MFS)

The maximum frame size which can pass over the channel. This parameter allows resource use of memory and processor to be shifted between the host and MDI or ICA-II.

The specified value is rounded up to be a multiple of 8. Default is 8512 bytes.

NOTE

If the interface is not the boot source, maximum frame size takes effect when the interface becomes operational. If the interface is the boot source, a reset is forced at the end of configuration and the maximum frame size takes effect after the next reset.

NETWORK_ID (NI)

A network ID used to form a unique network address for routing messages to the host through this DI or ICA. If the host has more than one channel connected to the same network front end or to different front ends, then this parameter must be specified. The unique network ID ensures that all messages sent to a specific host connection arrive at the DI or ICA processing that connection. If this parameter is not specified, the network address used for this channel is formed from the network ID specified with the NAM/VE definition information entered during NOS/VE configuration.

CHANNEL_TIMEOUT (CT)

Specifies how the MCI software handles host peripheral processor (PP) interrupts. This parameter is intended for use by customers during joint software development with Control Data. The following values are allowed:

Value	Description
TRUE	The MCI software detects when a host PP stops operating, downs the channel interface, and attempts recovery.
FALSE	The MCI software does not detect when a host PP stops operating. FALSE should be specified only for the debugging of host PP software.

Default is TRUE.

START (S)

Specifies whether or not the configured element should be started. Default is TRUE; started.

CDCNET does not currently support network operator commands to start, stop, or cancel a VE interface. Therefore, do not specify START=FALSE. There is no way to start a VE interface if START=FALSE is specified on DEFINE_VE_INTERFACE.

Responses

VE interface <interface_name> is defined.

VE interface <interface_name> is defined and started.

--ERROR-- VE interface <interface_name> is already defined.

--ERROR-- Interface slot may not be specified for an ICA II system.

--ERROR-- Interface slot <interface_slot> 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 interface slot must be specified.

--ERROR-- Network_id <network_id> is already defined.

--ERROR-- Specified CHANNEL board is not on.

--ERROR-- Specified CHANNEL board is already in use.

--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 required only in DI systems to be configured with a VE interface. This command is not required in systems booted over the channel running the VE interface. If this command is performed for the channel over which the device boot was received, it serves to define the NETWORK_ID parameter and redefine parameters INTERFACE_NAME, MAXIMUM_FRAME_SIZE, and CHANNEL_TIMEOUT. To redefine the VE interface over which the DI was booted, this command must be in the configuration file.

- If you use a value above 13000 for the MAXIMUM_FRAME_SIZE parameter, no error will be reported, but the PP driver hangs.

  You will get optimum throughput by setting the channel message size to a value that corresponds to optimum use of the other media in the communications device. For example, in an MDI connected to Ethernet, the Transport layer breaks a channel message into blocks containing 1412 bytes of user data. Optimum channel message sizes for this MDI are multiples of 1412 plus 40 bytes for channel protocol overhead. CDCNET modifies the value specified for MAXIMUM_FRAME_SIZE by rounding up to the next multiple of eight. Since this could result in an extra Ethernet block, the value specified should first be rounded down to the next lower multiple of eight. Recommended values for MAXIMUM_FRAME_SIZE are 5688, 7096, 8512, 9920, 11336, and 12744.

  Applications that use the TCP/IP protocols typically use messages of 8000 bytes. Throughput of these applications is improved somewhat when these messages fit in a single channel message. Therefore, any value above 8000 should improve the throughput of these applications.

  A key factor in selecting a maximum channel message size is the amount of memory available in the communications device. Use values above 5688 only if the device contains three megabytes of memory or more.

Examples

```
define_ve_interface interface_slot=4 ..
interface_name=nos_ve_interface_1 start=no

VE interface nos_ve_interface_1 is defined.
```

DEFINE_X25_ASYNCTIP (DEFXA)
DI Configuration Procedure/NETOU Command

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 ASYNC TIP to terminals connected through asynchronous lines.

This command includes terminal usage parameters that, for lines controlled by the ASYNC 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(s)
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 *INFINITE*
CONNECTION_DISCONNECT_TIMEOUT = 0..1000 or *INFINITE*
USER_CONNECTION_LIMIT = 1..16
ACCEPT_REVERSE_CHARGES = boolean
START = boolean
PERMANENT_VIRTUAL_CIRCUITS = list 1..64 range of 1..4095
PVC_CHANNEL_CONCATENATION = boolean
PVC_PACKET_SIZE = 16..1024
PVC_WINDOW_SIZE = 1..127
VALIDATE_USERS = 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 are executed when a virtual circuit becomes active. Default is that no TDP is 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 are executed each time a virtual circuit connection to a terminal becomes active. This parameter is ignored if a TDP (see TDP parameter) is specified. Default is that no TUP is automatically executed for the virtual circuit.

PROCEDURE_FILE_OPTION (PFO)

This parameter allows for additional options for configuring switched virtual circuit terminal devices and attributes. The following keyword values are allowed:

Keyword Value	Description
LOGICAL_CHANNEL_ CONCATENATION (LCC)	Appends the X.25 logical channel number (LCN) to the TDP's or TUP's file name. For example, TDP=TDP_PDN_FILE becomes TDP_PDN_FILE_*lcn*.
CALLED_DTE_ CONCATENATION (CDC)	Appends the called DTE address to the TDP's or TUP's file name. For example, TDP=TDP_PDN_FILE becomes TDP_ PDN_FILE_called_dte_address.
CALL_DATA_PROCEDURE (CDP)	Selects the option to treat call data information as if it were a TDP file name.

Default is that the specified TDP or TUP file is executed for all switched virtual circuits (SVCs). That is, no concatenation of LCN or called DTE address and no procedure filename is in the call data 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 CALLED_DTE_ CONCATENATION. Default is that the entire DTE address is used.

CONNECTION_CONNECT_TIMEOUT (CCT)

Defines how much time the terminal user has to create the first $INPUT/$OUTPUT connection. The range is 20 through 1000 seconds. If no connection is established within that time, the virtual circuit is cleared. It may be necessary to increase this value if the VALIDATE_USERS parameter is TRUE. The keyword INFINITE indicates an infinite time.

The timer has a precision of +/- 2 seconds. Default is 120 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 is 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. Default is FALSE; 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. Default is TRUE; started.

PERMANENT_VIRTUAL_CIRCUITS (PVC)

Specifies the permanent virtual circuit logical channel numbers (in CCITT terminology, the logical channel group number and the logical channel number) the X.25 Asynchronous TIP supports. The maximum number of PVCs per trunk name is restricted to 64.

PVCs may be listed individually, or as a range, for example, PVC=(1,2,3,4,5,11..16) or PVC=(1..5,11..16), provided the total number of PVCs does not exceed the maximum 64 logical channels.

The logical channel numbers used must be within the range of the PVC_RANGE parameter value on the DEFINE_X25_INTERFACE command.

This parameter is used in conjunction with the PVC_CHANNEL_CONCATENATION, PVC_PACKET_SIZE, and PVC_WINDOW_SIZE parameters.

PVC_CHANNEL_CONCATENATION (PCC)

This parameter is used in conjunction with the preceding PERMANENT_VIRTUAL_CIRCUITS parameter, and is only significant when PVCs have been defined. This parameter controls whether logical channel concatenation is to be performed on PVC logical channels. Default is FALSE.

If a TDP or TUP file name is specified, and PCC is TRUE, the PVC logical channel number is appended to the TDP or TUP file name; for example, TDP_PDN_FILE_*lcn* or TUP_PDN_FILE_*lcn*.

PVC_PACKET_SIZE (PPS)

This parameter is used in conjunction with the preceding PERMANENT_VIRTUAL_CIRCUITS parameter, and is only significant when PVCs have been defined. This parameter defines the default packet size to be used by all defined PVC channels.

If specified, this parameter overrides the default packet size parameter value (DEFAULT_PACKET_SIZE parameter on the DEFINE_X25_INTERFACE command) when permanent virtual circuits are defined, and this packet size is used on all permanent virtual circuits. Default is the DEFAULT_PACKET_SIZE value on the DEFINE_X25_INTERFACE command.

PVC_WINDOW_SIZE (PWS)

This parameter is used in conjunction with the preceding PERMANENT_VIRTUAL_CIRCUITS parameter, and is only significant when PVCs have been defined. The parameter value defines the window size to be used by ALL defined PVC channels.

If specified, this parameter value overrides the default window size parameter value (DEFAULT_WINDOW_SIZE parameter on the DEFINE_X25_INTERFACE command) when permanent virtual circuits are defined, and is used on all permanent virtual circuits. Default is the DEFAULT_WINDOW_SIZE value on the DEFINE_X25_INTERFACE command.

VALIDATE_USERS (VU)

Specifies if users of this X.25 AsyncTip are validated when the validation is enabled by the DEFINE_USER_VALIDATION command. If TRUE is specified, validation is enabled. Default is TRUE.

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.

--ERROR-- The number of supported PVC channels per trunk is restricted to <nn>.

--ERROR-- PVC_WINDOW_SIZE is greater than 7 and DEFXI PSN parmeter is NORMAL for trunk <trunk_name>.

--ERROR-- Logical channel number <nn> is not defined (DEFXI command, PR parameter) for trunk <trunk_name>.

--ERROR-- Logical channel number <nn> is already assigned to another user for trunk <trunk_name>.

--FATAL-- Not enough memory available for required table space.

Examples

```
define_x25_asynctip trunk_name=x25_telenet

X.25 AsyncTip support defined for specified trunks.
```

DEFINE_X25_GW (DEFXG)
DI Configuration Procedure/NETOU Command

Purpose

This command, along with the ADD_X25_GW_INCALL (ADDXGI) and ADD_X25_GW_OUTCALL (ADDXGO) commands, defines 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 identifiers for which incoming calls are accepted by the gateway for these trunks. Two parameters specify the protocol identifiers, one for calls to be routed to NOS/VE systems or other gateways, and one for calls to be routed to NOS systems. 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_INCALL and ADD_X25_GW_OUTCALL command descriptions in this manual.

Format

DEFINE_X25_GW
GATEWAY_NAME = name
TRUNK_NAME = list 1..32 of name(s)
NOS_PROTOCOL_ID = list 1..7 of 2..255
CDCNET_PROTOCOL_ID or *VE_PROTOCOL_ID = list 1..7 of 2..255*
NOS_PORT_NUMBER = list 1..15 of 1..OFC(16)
VE_OUTCALL_TITLE = list 1..15 of name(s)
START = boolean
DTE_ADDRESS_PROTOCOL_ID = list 1..7 of 0..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). A NOS protocol ID must be unique for an X.25 trunk.

CDCNET_PROTOCOL_ID (CPI) or *VE_PROTOCOL_ID (VPI)*

Specifies one or more protocol IDs that identify incoming calls to be routed to NOS/VE systems or other gateways by application title. Default is C2(16). A CDCNET protocol ID must be unique for an X.25 trunk.

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 NDL OUTCALL statements have a PORT parameter value not equal to 0 (that is, any value but zero), then that PORT number must match the value of the NOS_PORT_NUMBER parameter on an X.25 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. Default is no NOS/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. Default is TRUE; started.

DTE_ADDRESS_PROTOCOL_ID (DAPI)

Specifies one or more protocol IDs that identify incoming calls to be routed to CDNA server applications, by constructing a title from the protocol ID and the called DTE address. If incoming calls contain no protocol ID, this parameter should be set to 0, because the DTE address is the only means to identify the called application in that case. The default is that the protocol ID and the DTE address, by themselves, cannot identify the called application. A DTE address protocol ID must be unique for an X.25 trunk.

You can use the title generated here to connect to a VE_OUTCALL_TITLE at another X.25 gateway. The format of the title is 'DTE_xx_yyyy' where *xx* is the protocol ID and *yyyy* is the called DTE address.

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.

Remarks

If CONS has been defined and started, X.25 Packet Level sends all incoming calls with no user data to CONS. If CONS has not been defined and started, X.25 Packet Level sends all incoming calls with no user data to the user that has defined a protocol ID of 0.

Examples

```
define_x25_gw gateway_name=x25_gw trunk_name=X25TELENET,..
nos_port_number=05(16)

X.25 gateway x25_gw is defined.
```

DEFINE_X25_INTERFACE (DEFXI)
DI Configuration Procedure/NETOU Command

Purpose

Defines the Packet Level parameters of the X.25 interface. 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.

This command is normally used with a DEFINE_X25_TRUNK command. It can also be used with a DEFINE_LLC_2_TRUNK command. When the peer LLC 2 entity resides in a CDCNET device, static configuration is normally not needed. The DEFINE_LLC_2_TRUNK and DEFINE_X25_INTERFACE commands are simulated internally in the CDCNET device. However, when the peer LLC 2 entity resides in a non-CDCNET device, the DEFINE_LLC_2_TRUNK and DEFINE_X25_INTERFACE commands are needed, especially in defining the X.25 virtual circuit ranges.

Several parameter values on this command must match the values assigned in your public data network (PDN) subscription. Review your PDN subscription values before using this command.

Format

DEFINE_X25_INTERFACE (DEFXI)
TRUNK_NAME = name
PUBLIC_DATA_NETWORK = 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 = 16..4096
CONGESTED_THRESHOLD = 12..255
START = boolean
INTERFACE_STANDARD = CCITT_1980 or *CCITT_1984*
SUBSCRIBED_FACILITIES = list 1..2 of keyword
VALIDATE_LDA = 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 or the DEFINE_LLC_2_TRUNK command.

PUBLIC_DATA_NETWORK (PDN)

Specifies the name of the PDN for trunks connected to X.25 PDNs. Allowed keyword values include the following:

TELENET
TYMNET
UNINET
CDSN
DATAPAC
TRANSPAC
USERPSN1
USERPSN2
USERPSN3
USERPSN4
PSS

For X.25 trunks connected to the PDNs TELENET or DATAPAC, you must specify the keywords TELENET or DATAPAC to allow proper interoperation with these PDNs. For X.25 trunks that are not connected to PDNs, you can use this parameter to supply trunk subtype information (for example, WATS or INWATS). The value of this parameter is reported in all accounting messages related to this X.25 interface.

INTERFACE_NAME (IN)

Logical name of the X.25 interface. This name is used to see the X.25 interface in subsequent commands. Default is 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.

CONS puts this given address in the calling DTE field for all call requests going out over this X.25 interface (except when operating over an LLC 2 trunk, in which case, null DTE addresses are the default). However, CONS can be configured (see ADD_CONS_OUTCALL) with a LOCAL_DTE_ ADDRESS that can override the value configured by this command.

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 using modulo 8.
EXTENDED	Specifies extended packet numbering performed using modulo 128.

Default is NORMAL.

The PSN must be coordinated with the optional user facilities selected in your PDN subscription.

PVC_RANGE (PR)

Specifies the range of logical channel numbers (in CCITT terminology, the logical channel group number and the logical channel number) to be used for permanent virtual circuits. This parameter must match your PDN subscription value. This parameter is ignored if the specified trunk was defined by DEFINE_LLC_2_TRUNK (PVCs are not supported for LLC 2 trunks).

INONLY_RANGE (IR)

Specifies the range of logical channel numbers (in CCITT terminology, the logical channel group number and the logical channel number) to be used for incoming calls only. This parameter value must match your PDN subscription value.

TWOWAY_RANGE (TR)

Specifies the range of logical channel numbers (in CCITT terminology, the logical channel group number and the logical channel number) to be used for either incoming or outgoing calls. This parameter value must match your PDN subscription value.

OUTONLY_RANGE (OR)

Specifies the range of logical channel numbers (in CCITT terminology, the logical channel group number and the logical channel number) to be used for outgoing calls only. This parameter value must match your PDN subscription 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 is 2. This parameter value must match your PDN subscription value.

DEFAULT_PACKET_SIZE (DPS)

Specifies the data packet size, in bytes, 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 for CCITT_1980.

16
32
64
128
256
512
1024

Additional values of 2048 and 4096 are allowed for CCITT_1984. This parameter value must match your PDN subscription value. Default is 128.

Value of this parameter must be coordinated with the maximum frame size configured for the underlying trunk. It must be smaller than the value of the maximum frame size for the trunk by at least 4 bytes.

The value should be specified only if the nonstandard default packet sizes facility has been subscribed for this interface.

CONGESTED_THRESHOLD (CT)

Specifies the number of messages in the interface outgoing queue at which the interface is considered congested. Default is 12.

START (S)

Specifies whether or not the configured element should be started. Default is TRUE; started.

INTERFACE_STANDARD (IS)

Specifies the X.25 standard to be used for this interface. CCITT_1980 specifies that the 1980 version of the CCITT X.25 recommendation should be used. Annex A of ISO 8878 is not supported when running the CCITT 1980 X.25 standard.

This parameter is ignored if the specified trunk was defined by DEFINE_LLC_2_TRUNK. CCITT_1984 is always used for LLC 2 trunks.

SUBSCRIBED_FACILITIES (SF)

Specifies the X.25 facilities which have been subscribed for this interface. The value is specified as a list of one or two keywords from the following list.

FLOW_CONT_PARAMETER_NEGOTIATION (FCPN)
NONSTAND_DEFAULT_PACKET_SIZES (NDPS)
NONE
ALL

Default is NONE. However, if a value is specified for the DPS parameter of this command, it is assumed that the *nonstandard default packet sizes* facility has been subscribed. Then the default value is changed to the keyword NONSTAND_DEFAULT_PACKET_SIZES.

VALIDATE_LDA (VL)

Specifies whether the incoming packets should have the local DTE address validated. This parameter exists only for OSI conformance tests. During testing, the local DTE address must be validated. Default is NO.

Remarks

X.25 packet level users can negotiate the maximum packet size for each virtual circuit if the *flow control parameter negotiation* facility has been subscribed. At present, only the OSI Connection mode network service (CONS) software takes advantage of this facility. CONS uses the maximum frame size for the underlying trunk for this negotiation.

If the *nonstandard default packet sizes* facility has been subscribed, then a value can be specified for the DEFAULT_PACKET_SIZE parameter of this command.

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.

--ERROR-- Default packet size must be at least 4 bytes smaller than the maximum frame size for the underlying X.25 trunk <trunk_name>.

--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

X.25 interface X25TELENET defined.
```

DEFINE_X25_NET (DEFXN)
DI Configuration Procedure/NETOU Command

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..0FFFF(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
INCOMING_SERVICE_ACCESS = list 1..11 of keyword value
OUTGOING_SERVICE_ACCESS = list 1..11 of keyword value

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_xxxxx*, where *xxxxx* is the network ID expressed in decimal. For example, a network ID of 200 results in a default name of $NET_200.

COST (C)

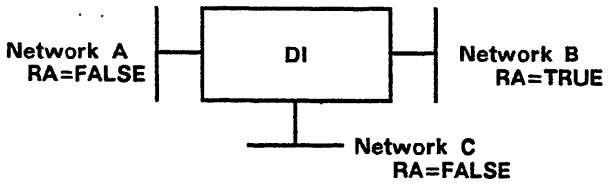
The relative cost of the X.25 network solution as a path for routing data through the network. The cost of a network may be calculated by dividing 100 million by the data rate of the network in bits per second. For example, the cost of a trunk with a speed of 56000 bits per second is 06FA(16). If the cost specified is greater than 0FFFF(16), 0FFFF(16) is used for the value of the COST parameter for compatibility reasons. If the trunk speed specified on the DEFINE_X25_TRUNK command is 1200, default is 0FFFF(16).

CDCNET routing uses cost to determine the least-cost routes for interconnecting networks. If multiple paths are available, the least-cost path is used. As a network becomes congested, its cost increases.

RELAY_ALLOWED (RA)

Specifies how the DI may use this network for the relaying of traffic through the DI. Default for an X.25 network solution is TRUE. If RA is TRUE, the DI relays traffic between this network and other connected networks regardless of the definition of the other networks. If RA is FALSE, the DI still relays traffic between this network and any connected networks defined with RA=TRUE. If RA is FALSE, the DI system does not relay traffic between this network and other connected networks defined with RA=FALSE *unless* no other nonrestricted route exists between the networks.

The following example illustrates these rules.



Relay Between:	Allowed/Not Allowed:
Network A and Network B	Allowed (RA is TRUE for Network B).
Network A and Network C	Not allowed (RA is FALSE for both Network A and Network C).
Network B and Network C	Allowed (RA is TRUE for Network B).

ROUTING_INFO_NETWORK (RIN)

Specifies whether or not the network solution is to carry CDCNET routing information. Routing information pertains both to the proprietary CDCNET IS-IS routing protocol, as well as the ES-IS routing protocol. 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 appears, then, as not connected to any network other than this network solution. Default 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. Default 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. Default is TRUE; system accepts PDN charges.

START (S)

Specifies whether or not the X.25 network solution should be started after it is configured. Default is TRUE; started.

ARCHITECTURE_TYPE (AT)

This parameter is no longer supported. Specification is valid but the value is ignored.

OUTPUT_QUEUE_LIMIT (OQL)

Maximum data, in bytes, retained in the output queue for the network solution if the DI's operating system buffer queue state is poor or worse. If messages must be discarded, newest output messages are discarded first.

Default 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, default is 60000 bytes.

INCOMING_SERVICE_ACCESS (ISA)

Specifies which services may be accessed coming in via the network solution. The keyword values are:

Keyword Value	Description
ALL	All services are allowed.
NONE	No services are allowed.
application	Any service not identifiable by keyword batch_support, interactive, file_transfer, or network_management.
batch_support	Any batch input and output services are allowed. This includes batch control facility, batch job submittal, and batch output.

Keyword Value	Description
interactive	All interactive services are allowed. Interactive service supports terminal user access. Specifying this keyword is the same as specifying all of the following interactive service keywords: nos_interactive, passthrough, t_passthrough, user_telnet, x25_terminal_gw, and ve_interactive.
nos_interactive	All interactive services provided by NOS mainframes are allowed.
passthrough	All site-defined passthrough services are allowed. A site-defined passthrough service is provided when a passthrough title is created during the execution of a Terminal User Procedure (TUP) when a line is activated. This service allows asynchronous interactive devices to make connections to passthrough titles of this type through asynchronous ports and exchange information using CDCNET's transparent data feature.
t_passthrough	All user-defined passthrough services are allowed. A user-defined passthrough service is provided when an interactive terminal user creates a connection to the interactive passthrough server, and defines a passthrough title either by entering the required commands manually or by executing a Terminal User Procedure (TUP) via the DO command on the $CDCNET_COMMAND connection. This service allows asynchronous interactive devices to make connections to passthrough services of this type through asynchronous ports and to exchange information using CDCNET's transparent data feature.
user_telnet	All TELNET services are allowed. User TELNET is an interactive service which provides CDCNET terminal users with access to the interactive services of remote hosts on a TCP/IP network.
x25_terminal_gw	All X.25 terminal services are allowed. The X.25 terminal gateway is an interactive service which provides CDCNET asynchronous terminal users with access to the interactive services of remote hosts on an X.25 network.

Keyword Value	Description
ve_interactive	All interactive services provided by NOS/VE mainframes are allowed.
file_transfer	All QTF/PTF file transfer services are allowed. File transfer services support user file (PTF) and queue file transfers (QTF).
network_management	All network alarm, clock, command, file, and log services are allowed.

Default is ALL.

OUTGOING_SERVICE_ACCESS (OSA)

Specifies which services may be accessed going out via the network solution. See the ISA parameter for the keyword values.

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=X25TELNET,..
remote_dte_address='6124821234',network_name=X25NET,network_id=0a(16)

X25 network X25NET defined for trunk X25TELNET.
```

DEFINE_X25_TERMINAL_GW (DEFXTG)
DI Configuration Procedure/NETOU Command

Purpose Defines the X.25 terminal gateway service for one or more X.25 trunks. It establishes a logical association between the trunks so the trunks can be viewed as one logical link to the PDN.

DEFXTG also defines a CDCNET title, which is used by CDCNET asynchronous terminal users to connect to the gateway in order to access applications available in an X.25 network.

Format

DEFINE_X25_TERMINAL_GW
GATEWAY_NAME = name
TRUNK_NAME = list 1..32 of name
TITLE = name
OUTCALL_DEFINITION_REQUIRED = boolean
START = boolean

Parameters

GATEWAY_NAME (GN)

Specifies the logical name of the gateway used in subsequent commands which reference the gateway.

TRUNK_NAME (TN)

Specifies the logical name(s) 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, then a separate gateway definition must be specified for each PDN.

TITLE (T)

Specifies a title that the X.25 terminal gateway registers. This allows terminal users to create a connection to the X.25 terminal gateway. It interactively enters the DTE address of the foreign application or selects a preconfigured DTE address. The default is the value supplied for the GATEWAY_NAME parameter.

OUTCALL_DEFINITION_REQUIRED (ODR)

Specifies whether or not outcalls (see the ADDXTGO command in this manual) must be defined by the network administrator before the X.25 terminal manual gateway can become fully operational. It allows the network administrator to control or restrict terminal users access/use of the X.25 terminal gateway.

If this parameter is TRUE, the terminal user may establish a foreign connection only to a DTE address that matches a DTE address entered by the network administrator on the ADDXTGO command. The title is not registered until an ADDXTGO has been performed. If this parameter is FALSE (outcalls not needed), unrestricted access is allowed. Default is FALSE.

START (S)

Specifies whether or not the gateway should be started. The default value is TRUE (started).

Responses

X.25 terminal gateway <gateway_name> is defined.

X.25 terminal gateway <gateway_name> is defined and started.

X.25 terminal gateway <gateway_name> is defined and awaiting outcall definition.

--ERROR-- X.25 terminal gateway <gateway_name> is already defined.

--ERROR-- The specified X.25 trunks do not connect to the same PDN.

--ERROR-- Trunk name <trunk_name> is not an X.25 trunk.

--ERROR-- Trunk name <trunk_name> may not appear more than once in the list of specified trunks.

--ERROR-- X.25 terminal gateway title <title> is already defined.

--FATAL-- X.25 terminal gateway - not enough memory available for the required table space.

Examples

This example allows any asynchronous terminal user in the CDCNET network to connect to the DATAPAC title and call any X.25 remote application connected to the DATAPAC PDN.

```
define_x25_terminal_gw gateway_name=datapac_gateway ..
trunk_name=datapac_trunk ..
title=datapac ..
outcall_definition_required=FALSE ..
start=TRUE

X.25 terminal gateway datapac_gateway is defined and
started.
```

DEFINE_X25_TRUNK (DEFXT)
DI Configuration Procedure/NETOU Command

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 = 1..9
DRIVER = keyword value
MAX_FRAME_SIZE = 132..32768

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 operates as the Data Communications Equipment end for the X.25 trunk.
DTE	Specifies that the DI operates as the Data Terminating Equipment end for the trunk.

Default 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 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). Default is 500 milliseconds.

The following table shows the optimum values for PF_RECOVERY_ TIMER, based on trunk speed.

Trunk Speed	PRT (milliseconds)
1200	32000
2400	16000
4800	8000
9600	4000
19200	2000
38400	1000
48000	750
56000	500
64000	500
128000	500
256000	500

RETRANSMISSION_LIMIT (RL)

Maximum retransmissions allowed. Default is 5 retransmissions.

TRUNK_SPEED (TS)

Speed of the X.25 trunk in bits per second. The LIM uses trunk speed to generate data clocking for the trunk (except when clocking is external), and to configure the trunk with proper values for network cost and output queue limit. Trunk speed is also used by NETCU to set the value for the PF_RECOVERY_TIMER ERROR_RECOVERY_TIMER parameter (see table in PF_RECOVERY_TIMER parameter description). The possible values for this parameter are:

1200
2400
4800
9600
19200
38400
48000
56000
64000
128000
256000

Default is 56000. Specifying a trunk speed that differs from the actual speed, even if you have specified EXTERNAL clocking, results 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	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 RS-232 DD circuit for RS-232 ports, or the RS-449 SR circuit for RS-449 ports. The EXTERNAL transmit data is derived from the RS-232 DB circuit for RS-232 ports, or the RS-449 ST circuit for RS-449 ports.
INTERNAL	Specifies that the LIM generates the clocking for both transmit and receive data. A single clock signal is generated; it matches the TRUNK_SPEED specified for the trunk. The LIM supplies the clock on the RS-232 DA or RS-449 TT circuit.
TRANSMIT	The LIM generates clocking for transmit data, but derives 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 RS-232 DA circuit (for RS-232 ports) or the RS-449 TT circuit (for RS-449 ports). The LIM derives the receive data clock from the RS-232 DD circuit (for RS-232 ports) or the RS-449 SR circuit (for RS-449 ports).

Default is EXTERNAL.

Clocking should be INTERNAL or 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. Default is 7. For example, a value of 7 on this parameter results, on average, in 70 bytes of interactive priority data for every 30 bytes of batch priority data.

DRIVER (D)

Specifies the type of HDLC driver used for this X.25 trunk. The following keyword values are allowed:

Keyword Value	Description
HDLC_NORMAL	Operates with other line types on the same CIM. It operates one HDLC line at 64 K bits per second and a combination of HDLC lines not exceeding 22,000 characters per second.
HDLC256	Operates one HDLC line at 128 K bits per second or 256 K bits per second on a dedicated CIM.
HDLC3x64	Operates one to three HDLC lines at speeds up to 64 K bits per second per line on a dedicated CIM.

Default is HDLC_NORMAL.

MAX_FRAME_SIZE (MFS)

Specifies the maximum frame size of the HDLC frame which may be transmitted.

The value of this parameter must be coordinated with the value of the MAX_DATAGRAM_SIZE parameter in the DEFINE_IP_NET command or the DEFAULT_PACKET_SIZE parameter in the DEFINE_X25_INTERFACE command if these commands configure an IP network or X.25 interface over this trunk. Default is 1500.

Responses

X.25 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-- Line_speed <integer> is not supported for an X.25 trunk.

--ERROR-- X.25 is not supported on the specified LIM.

--ERROR-- A line or trunk that is already defined is serviced by the same CIM as this trunk. The HDLC256 driver requires the CIM to be dedicated to this trunk.

--ERROR-- A line that is already defined is serviced by the same CIM as this trunk. The HDLC3X64 driver requires the CIM to be dedicated to HDLC or X.25 trunks.

--ERROR-- A trunk that is already defined for driver <driver> is serviced by the same CIM as this trunk. A CIM can only support one HDLC driver.

--ERROR-- Three HDLC or X.25 trunks that are already defined are serviced by the same CIM as this trunk. The hdlc3X64 driver cannot support more than three trunks.

--ERROR-- The DI was loaded via this HDLC trunk and the CIM switch setting specified the hdlc256 driver. The driver cannot be changed to hdlc_ normal or hdlc3x64 since these drivers do not handle high line speeds and communication could be lost.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_x25_trunk lim=0 port=0 trunk_name=X25TELNET

X25 trunk X25TELNET defined.
```

DELETE Commands 7

DELETE Commands 7

This chapter provides complete descriptions of all CDCNET Network Operations DELETE commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

DELETE_ARP_REPORTED_NETS (DELARN) NETOU Command

Purpose Deletes an IP reported network entry for the specific source IP network.

Format

DELETE_ARP_REPORTED_NETS
SOURCE_IP_NETWORK = list 1..4 of 0..255

Parameters

SOURCE_IP_NETWORK (SIN)

Specifies the network or subnetwork from which ARP requests are received for other networks. The DI is no longer restricted from sending ARP replies to the specified source IP network, from which ARP requests are received for other IP networks. The DI can now respond to all ARP requests from the source IP network.

Responses Reportable IP network/subnet is deleted for source IP network/subnet <source_ip_network>.

--ERROR-- Reportable IP network/subnet is not defined for source IP network/subnet <source_ip_network>.

Remarks The ADD_ARP_REPORTED_NETS command supplies a list of destination IP networks. Any ARP request to the DI from the SOURCE_IP_NETWORK are only responded to by the DI if the IP network is on that destination list defined in the ADD_ARP_REPORTED_NETS command. The DELETE_ARP_REPORTED_NETS command takes off the destination restrictions and allows the DI to respond to all ARP requests from the SOURCE_IP_NETWORK.

Examples

```
delete_arp_reported_nets source_ip_network = (192,5,209)

Reportable IP network/subnet is deleted for source IP
network/subnet 192.5.209.
```

DELETE_CLNS_FILTER (DELCF)
NETOU Command

Purpose Deletes a CLNS filter entry, previously defined using the ADD_CLNS_FILTER command. To delete a CLNS filter, both address prefix parameters from the command must exactly match the CLNS filter entry. If ALL (default value) is specified for both of the address prefix parameters, all CLNS filter entries for the specified network are deleted. You cannot specify ALL for only one of the address prefix parameters.

Format

DELETE_CLNS_FILTER
NETWORK_NAME = name
SOURCE_ADDRESS_PREFIX = string 2..40 or *ANY* or *ALL*
DESTINATION_ADDRESS_PREFIX = string 2..40 or *ANY* or *ALL*

Parameters

NETWORK_NAME (NN)

Identifies the name of a currently defined network within this system. This is the network on which the CLNS filter entry to be deleted is defined.

SOURCE_ADDRESS_PREFIX (SAP)

Specifies the NSAP address prefix of the CLNS filter entry to delete. The keyword ANY matches a CLNS filter entry whose SOURCE_ADDRESS_PREFIX parameter is ANY. If ALL is specifed for both of the address prefix parameters, all CLNS filter entries for the specified network are deleted. An address prefix must include an even number of hexadecimal digits. The default is ALL.

DESTINATION_ADDRESS_PREFIX (DAP)

Specifies the NSAP address prefix of the CLNS filter entry to delete. The keyword ANY matches a CLNS filter entry whose DESTINATION_ADDRESS_PREFIX is ANY. If ALL is specifed for both of the address prefix parameters, all CLNS filter entries for the specified network are deleted. An address prefix must include an even number of hexadecimal digits. The default is ALL.

Responses

The specified CLNS filter entry is deleted from network <network_name>.

All CLNS filter enties are deleted from network <network_name>.

--WARNING-- No CLNS filter entires are defined for network <network_name>.

--ERROR-- Network <network_name> is not defined.

--ERROR-- The specified CLNS filter entry is not defined on network <network_name>.

--ERROR-- ALL cannot be specified for just one of the address prefix parameters.

--ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

--FATAL-- Not enough memory available for required table space.

Examples

```
delete_clns_filter network_name=NETWORK1 source_address_prefix='044723FF' ..
 destination_address_prefix='0032004556'

 The specified CLNS filter entry is deleted from network NETWORK1.


delete_clns_filter network_name=NETWORK1 ..
  source_address_prefix=all destination_address_prefix=all

 All CLNS filter entries are deleted from network NETWORK1.


delete_clns_filter nn=NETWORK1 sap='044723FF' dap='0032004556'

 --ERROR-- The specified CLNS filter entry is not defined for network NETWORK1.


delete_clns_filter nn=NETWORK2 sap='044723FF' dap='0032004556'

 --ERROR-- Network NETWORK2 is not defined.


delete_clns_filter nn=NETWORK1 sap='044723FF' dap=ALL

 --ERROR-- ALL cannot be specified for just one of the address prefix parameters.


 delete_clns_filter nn=NETWORK1 sap='044EEFHH' dap='0032004556'

 --ERROR-- The NSAP address prefix 044EEFHH contains non-hexadecimal characters.
```

DELETE_CONS_INCALL (DELCI)
NETOU Command

Purpose Deletes one or more previously configured CONS incall definitions added by the ADD_CONS_INCALL command.

Format

DELETE_CONS_INCALL
ADDRESS_PREFIX = list 1..10 of string 2..40 or ALL

Parameters

ADDRESS_PREFIX (AP)

Specifies the set of NSAP address prefixes associated with the CONS incall definition entries to be deleted. Use the value ALL to delete the entries for all addresses. The elements of each string defining an address prefix must be hexadecimal digits, unless you specify ALL.

Responses The following response is generated if the value of the address prefix parameter specifies a single address prefix:

> The CONS incall definition entry for the NSAP address prefix <address_prefix> is deleted.

The following response is generated if the value of the address prefix parameter specifies a list of two or more address prefixes:

> The CONS incall definition entries for the following NSAP address prefixes are deleted.
> {The above line is followed by a list of address prefixes. Please see the command response example for the format in which this information is displayed.}

The following response is generated if a value of ALL is specified for the address prefix parameter:

> CONS incall definition entries for all NSAP address prefixes are deleted.

The following response is generated if no address prefixes have been configured, regardless of input errors:

> --WARNING-- No CONS incall definition entry for any NSAP address prefix is defined.

The following response is generated if a requested address prefix has not been configured:

> --ERROR-- A CONS incall definition entry for the NSAP address prefix <address_prefix> is not defined.

The following response is generated if an ADDRESS_PREFIX parameter contains an odd number of digits:

> --ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

The following response is generated if an ADDRESS_PREFIX parameter contains a nonhexadecimal character.

> --ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

Examples

```
delete_cons_incall address_prefix='044723FF'

The CONS incall definition entry for NSAP address prefix 044723FF is deleted.

delete_cons_incall address_prefix=('044723FF', '44AAEEFF')

The CONS incall definition entries for the following NSAP address prefixes are deleted.

  044723FF
  44AAEEFF

delete_cons_incall address_prefix='all'

CONS incall definition entries for all NSAP address prefixes are deleted.

delete_cons_incall address_prefix='044EEFHH'

--ERROR-- The NSAP address prefix 044EEFHH contains non-hexadecimal characters.
```

DELETE_CONS_OUTCALL (DELCO)
NETOU Command

Purpose Deletes one or more previously configured CONS outcall definitions added by the ADD_CONS_OUTCALL command.

Format

DELETE_CONS_OUTCALL
ADDRESS_PREFIX = list 1..10 of string 2..40 or ALL

Parameters

ADDRESS_PREFIX (AP)

Specifies the set of NSAP address prefixes associated with the CONS outcall definition entries to be deleted. Use the value ALL to delete the entries for all CONS outcall definition entries for addresses. The elements of each string defining an address prefix must be hexadecimal digits, unless you specify ALL.

Responses The following response is generated if the value of the address prefix parameter specifies a single address prefix:

> The CONS outcall definition entry for the NSAP address prefix <address_prefix> is deleted.

The following response is generated if the value of the address prefix parameter specifies a list of two or more address prefixes:

> The CONS outcall definition entries for the following NSAP address prefixes are deleted.
> {The above line is followed by a list of address prefixes. Please see the command response example for the format in which this information is displayed.}

The following response is generated if a value of ALL is specified for the address prefix parameter:

> CONS outcall definition entries for all NSAP address prefixes are deleted.

The following response is generated if no address prefixes have been configured, regardless of input errors:

> --WARNING-- No CONS outcall definition entry for any NSAP address prefix is defined.

The following response is generated if a requested address prefix has not been configured:

> --ERROR-- A CONS outcall definition entry for the NSAP address prefix <address_prefix> is not defined.

The following response is generated if an ADDRESS_PREFIX parameter contains an odd number of digits:

> --ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

The following response is generated if an ADDRESS_PREFIX parameter contains a nonhexadecimal character:

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

Examples

```
delete_cons_outcall address_prefix='044723FF'

The CONS outcall definition entry for NSAP address prefix 044723FF is deleted.

delete_cons_outcall address_prefix=('044723FF', '44AAEEFF')

The CONS outcall definition entries for the following NSAP address prefixes are deleted.

  044723FF
  44AAEEFF

delete_cons_outcall address_prefix='all'

CONS outcall definition entries for all NSAP address prefixes are deleted.

delete_cons_outcall address_prefix='044EEFHH'

--ERROR-- The NSAP address prefix 044EEFHH contains non-hexadecimal characters.
```

DELETE_CONS_1980_X25_MAPPING (DELC1XM) NETOU Command

Purpose Deletes a previously configured NSAP address combination used to route incoming CONS/TP0 traffic received from a 1980 X.25 PDN. This command deletes routing information that was previously added via the ADD_CONS_1980_X25_MAPPING command.

Format

DELETE_CONS_1980_X25_MAPPING
DESTINATION_NSAP_ADDRESS = string 2..40 or ALL
DESTINATION_DTE_ADDRESS = string 1..15
SOURCE_NSAP_ADDRESS = string 2..40
SOURCE_DTE_ADDRESS = string 1..15

Parameters

DESTINATION_NSAP_ADDRESS (DNA)

Specifies the destination NSAP address that represents the routing entry to be deleted. The elements of the string entered must contain an even number of hexadecimal digits. If ALL is specified, all entered addresses are deleted.

DESTINATION_DTE_ADDRESS (DDA)

Specifies the destination DTE address that represents the routing entry to be deleted. The elements of the string entered must contain decimal digits. This parameter is required if the ADDC1XM information being deleted was used to define the DDA.

SOURCE_NSAP_ADDRESS (SNA)

Specifies the source NSAP address that represents the routing entry to be deleted. The elements of the string entered must contain an even number of hexadecimal digits. This parameter is required if the ADDC1XM information being deleted was used to define the SNA.

SOURCE_DTE_ADDRESS (SDA)

Specifies the source DTE address that represents the routing entry to be deleted. The elements of the string entered must contain decimal digits. This parameter is required if the ADDC1XM information being deleted was used to define the SDA.

Responses The following information for routing incoming CONS/TP0 traffic has been deleted:

Destination NSAP address = <destination_nsap_address>
Destination DTE address = <destination_dte_address>
Source NSAP address = <source_nsap_address>
Source DTE address = <source_dte_address>

All CONS 1980 X.25 routing information has been deleted.

--WARNING-- The given information for routing incoming CONS/TP0 traffic is not defined.

--WARNING-- No CONS 1980 X.25 routing information exists.

--ERROR-- The NSAP address <nsap address> contains non-hexadecimal characters.

--ERROR-- The NSAP address <nsap address> contains an odd number of digits.

--ERROR-- The DTE address <dte address> contains non-decimal characters.

Examples

```
delete_cons_1980_x25_mapping dda='1234567' ..
 dna='39840f01bb7b000000000000108002510008401'..
 sda='789012'  sna='47000400030001202301100700001'

 Destination NSAP address = 39840f01bb7b000000000000108002510008401
 Destination DTE address = 1234567
 Source NSAP address = 47000400030001202301100700001
 Source DTE address = 789012


delete_cons_1980_x25_mapping ..
 dna='39840f01bb7b000000000000108002510008401'

  Destination NSAP address = 39840f01bb7b000000000000108002510008401
  Destination DTE address = Not Specified
  Source NSAP address = 39840f01bb7b000000000000108002510007201
  Source DTE address = Not Specified


 delc1xm all

  All CONS 1980 X.25 routing information has been deleted.


 delc1xm  dda='1234567'  dna='0000000008002510008G'

 --ERROR-- The NSAP address 0000000008002510008G
contains non-hexadecimal characters.
```

DELETE_EGP_REPORTED_NET (DELERN) NETOU Command

Purpose Deletes the definition of an IP network address which EGP reports to its peers.

Format

DELETE_EGP_REPORTED_NET
IP_NETWORK = list 1..3 of 0..255 or keyword value

Parameters

IP_NETWORK (IN)

Specifies the IP network for which peer reporting is to be deleted. Specify an IP network number if the reporting on a specific IP network is to be deleted. The following keyword values are also allowed.

Keyword Value	Description
ALL	The entire list of reported local networks is deleted and EGP reverts to reporting all networks that the DI has learned.
NONE	The entire list of reported local networks is deleted and EGP stops reporting any network numbers.

Default is NONE.

Responses

EGP reported network <ip_network> is deleted.

EGP reported network deleted, ALL are reported.

EGP reported network deleted, NONE are reported.

--ERROR-- EGP network <ip_network> is not defined.

Examples

```
delete_egp_reported_net in = (128,5)

EGP reported network 128.5 is deleted.
```

DELETE_INCOMING_SERVICE_ACCESS (DELISA) NETOU Command

Purpose Deletes previously added titles of servers that may be accessed coming in via the network solution.

Format

DELETE_INCOMING_SERVICE_ACCESS
TITLE = list 1..15 of name or string 1..255
NETWORK NAME = name

Parameters

TITLE (T)

Specifies a title of a server that may be accessed coming in via the network solution. The title must match a previously added title exactly, including any wildcards.

Title is case-sensitive when entered as a string. For example, a title entered in uppercase with the ADDISA command must always be entered in uppercase on this command, or it is seen as a different title.

NETWORK_NAME (NN)

Identifies the name of a currently defined network within this system. This is the network solution upon which the accessible services are being defined. Default is the previously defined network when on the configuration file. It is **required** when entered by the operator.

Responses

Incoming service access deleted from network <network_name>.

--WARNING-- Incoming service access deleted from network <network_name>.
Title <title> not found.
{Line repeats for each title not found}

--ERROR-- A network is not defined.

--ERROR-- Network <network_name> is not defined.

--ERROR-- Title <title> may not appear more than once in the list of specified titles.

--ERROR-- Expecting NAME or STRING, found <xxx>.

Examples

```
delete_incoming_service_access title='ARH907' network_name=$NET_15

Incoming service access deleted from network $NET_15.
```

DELETE _IP_FILTER (DELIF)
NETOU Command

Purpose Deletes IP filter entries, previously defined using the ADD_IP_FILTER command. All IP address, IP mask, protocol, and port number parameters from the command must exactly match for an IP filter entry to be deleted. If either IP address parameter is set to ALL or not specified (defaults are ALL), other parameters are ignored and all IP filter entries for the specified IP network are deleted.

Format

DELETE _IP_FILTER
IP_NETWORK = list 1..4 of integer 0..255
SOURCE _IP_ADDRESS = list 1..4 of integer 0..255 or *ANY* or *ALL*
DESTINATION _IP_ADDRESS = list 1..4 of integer 0..255 or *ANY* or *ALL*
SOURCE _IP_MASK = list 1..4 of integer 0..255 or *HOST*
DESTINATION _IP_MASK = list 1..4 of integer 0..255 or *HOST*
PROTOCOL = integer of 1..254 or *keyword value*
SOURCE _PORT_NUMBER = list 1..10 of integer of 0..65535 or *ANY*
DESTINATION _PORT_NUMBER = list 1..10 integer of 0..65535 or *ANY*

Parameters

IP_NETWORK (IN)

Identifies an IP network currently defined within this system. This is the network on which the IP filter entry to be deleted is defined.

SOURCE _IP_ADDRESS (SIA)

Specifies the source IP address of the IP filter entry to be deleted. The keyword ANY matches an IP filter entry whose SOURCE _IP_ADDRESS is ANY. If ALL is specifed for both of the address prefix parameters, all IP filter entries for the specified network are deleted. The default is ALL.

DESTINATION _IP_ADDRESS (DIA)

Specifies the destination IP address of the IP filter entry to be deleted. The keyword ANY matches an IP filter entry whose DESTINATION _IP_ ADDRESS is ANY. If ALL is specifed for both of the address prefix parameters, all IP filter entries for the specified network are deleted. The default is ALL.

SOURCE _IP_MASK (SIM)

Specifies the source IP mask of the IP filter entry to be deleted. The keyword HOST matches an IP filter entry whose SOURCE _IP_MASK is HOST. The default is HOST.

DESTINATION _IP_MASK (DIM)

Specifies the destination IP mask of the IP filter entry to be deleted. The keyword HOST matches an IP filter entry whose DESTINATION _IP_ MASK is HOST. The default is HOST.

PROTOCOL (P)

Specifies the protocol of the IP filter entry to be deleted. The keywords ICMP, TCP, and UDP specify corresponding protocol values. The keyword ANY matches an IP filter entry whose PROTOCOL is ANY. The default is ANY.

SOURCE_PORT_NUMBER (SPN)

Specifies ranges of source port number values of the IP filter entry to be deleted. The keyword ANY matches an IP filter entry whose SOURCE_PORT_NUMBER is ANY. The default is ANY.

DESTINATION_PORT_NUMBER (DPN)

Specifies ranges of destination port number values of the IP filter entry to be deleted. The keyword ANY matches an IP filter entry whose DESTINATION_PORT_NUMBER is ANY. The default is ANY.

Responses

The specified IP filter entry is deleted from IP network <ip_network>.

All IP filter entries are deleted from IP network <ip_network>.

--WARNING-- No IP filter entries are defined on IP network <ip_network>.

--ERROR-- IP network <ip_network> is not defined.

--ERROR-- The specified IP filter entry is not defined on IP network <ip_network>.

Examples

```
delete_ip_filter ip_network=(129,179,52) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255

 The specified IP filter entry is deleted from IP network 129.179.52.


delete_ip_filter ip_network=(129,179,52)

 All IP filter entries are deleted from IP network 129,179,63.


delete_ip_filter ip_network=(129,179,52) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255

 --ERROR-- The specified IP filter entry is not defined on IP network
 129.179.52.


delete_ip_filter ip_network=(129,179,54) source_ip_address=(129,179) ..
  destination_ip_address=(129,179,63) source_ip_mask=(255,255) ..
  destination_ip_mask=host protocol=any source_port_number=1..126 ..
  destination_port_number=127..255

 --ERROR-- IP network 129.179.54 is not defined.
```

DELETE_NON_CDC_ADDRESS_PREFIX (DELNCAP) NETOU Command

Purpose Deletes a previously configured address prefix of non-CDCNET NSAP addresses, and information about the associated next hop.

Format

DELETE_NON_CDC_ADDRESS_PREFIX
ADDRESS_PREFIX = list 1..10 of string 2..40 or ALL

Parameters

ADDRESS_PREFIX (AP)

Specifies the set of address prefixes of non-CDCNET NSAP addresses which are to be deleted. Specify ALL if all address prefixes are to be deleted. Specify each address prefix in hexadecimal, and enclose each address prefix in single quotes.

Responses The following response is repeated for each address prefix affected by the command when you specify one or more valid address prefixes.

The NonCDCNET NSAP address prefix <address_prefix> is deleted.

The following response is displayed if you specify ALL for the address_prefix parameter.

All NonCDCNET NSAP address prefixes are deleted.

The following response is displayed if a requested address_prefix has not been configured.

--WARNING-- The NonCDCNET NSAP address prefix <address_prefix> is not defined.

The following response is displayed if no NonCDCNET address_prefixes have been configured.

--WARNING-- No NonCDCNET NSAP address prefix is defined.

The following response is displayed for each value of the address_prefix parameter which contains an odd number of digits.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains an odd number of digits.

The following response is displayed for each value of the address_prefix parameter which contains a non-hexadecimal character.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains non-hexadecimal characters.

The following response is displayed if the ES - IS routing is not defined.

--ERROR-- ES - IS routing is not defined.

Examples

```
delete_non_cdc_address_prefix ap='044723ff'

The NonCDCNET NSAP address prefix 044723FF is deleted.


delete_non_cdc_address_prefix ap=('044723ff', '44aaeeff')

The NonCDCNET NSAP address prefix 044723FF is deleted.
The NonCDCNET NSAP address prefix 44aaeeff is deleted.


delete_non_cdc_address_prefix ap='all'

All NonCDCNET NSAP address prefixes are deleted.


delete_non_cdc_address_prefix ap='044eefhh'

--ERROR-- The NonCDCNET NSAP address prefix '044eefhh' contains
non-hexadecimal characters.
```

DELETE_NON_CDC_ROUTING_ENTRY (DELNCRE) NETOU Command

Purpose Deletes a previously configured routing entry, when executed in the same boundary system in which the routing entry was added, by using the ADD_NON_CDC_ROUTING_ENTRY command.

Format

DELETE_NON_CDC_ROUTING_ENTRY
ADDRESS_PREFIX = list 1..10 of string 2..40 or ALL

Parameters

ADDRESS_PREFIX (AP)

Specifies the address prefix of non-CDC NSAP addresses for which the routing entry is to be deleted. The value ALL deletes all non-CDCNET NSAP addresses. The elements of each string must be hexadecimal digits.

Responses The following response is generated if the value of the address prefix parameter specifies a single address prefix.

The routing entry for the NonCDCNET NSAP address prefix <address_prefix> is deleted.

The following response is generated if the value of the address prefix parameter specifies a list of two or more address prefixes.

The routing entries for the following NonCDCNET NSAP address prefixes are deleted (see examples for list of entries deleted).

The following response is generated if a value of ALL is specified for the address prefix parameter.

Routing entries for all NonCDCNET NSAP address prefixes are deleted.

The following response is generated if a requested address prefix has not been configured.

--WARNING-- A routing entry for the NonCDCNET NSAP address prefix <address_prefix> is not defined.

The following response is generated if no NonCDCNET address prefixes have been configured.

--WARNING-- No routing entry for NonCDCNET NSAP address prefix is defined.

The following response is generated for each value of the address_prefix parameter which contains an odd number of digits.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains an odd number of digits.

The following response is generated for each value of the address_prefix parameter which contains a non-hexadecimal character.

--ERROR-- The NonCDCNET NSAP address prefix <address_prefix> contains non-hexadecimal characters.

Examples

```
delete_non_cdc_routing_entry ap='044723FF'

The routing entry for NonCDCNET NSAP address prefix 044723FF is deleted.


delete_non_cdc_routing_entry ap=('044723FF', '44AAEEFF')

The routing entries for the following NonCDCNET NSAP address prefixes
are deleted.
044723FF
44AAEEFF


delete_non_cdc_routing_entry ap='all'

Routing entries for all NonCDCNET NSAP address prefixes are deleted.


delete_non_cdc_routing_entry ap='044EEFHH'

--ERROR-- The NonCDCNET NSAP address prefix 044EEFHH contains
          non-hexadecimal characters.
```

DELETE_OUTGOING_SERVICE_ACCESS (DELOSA) NETOU Command

Purpose Deletes previously added titles of servers that may be accessed going out via the network solution.

Format

DELETE_OUTGOING_SERVICE_ACCESS
TITLE = list 1..15 of name or string 1..255
NETWORK NAME = name

Parameters

TITLE (T)

Specifies a title of a server that may be accessed coming in via the network solution. The title must match a previously added title exactly, including any wildcard characters (see the CDCNET Network Operations and Analysis manual for more information).

Title is case-sensitive when entered as a string. For example, a title entered in uppercase with the ADDOSA command must always be entered in uppercase on this command, or it is seen as a different title.

NETWORK_NAME (NN)

Identifies the name of a currently defined network within this system. This is the network solution upon which the accessible services are being defined. Default is the previously defined network when on the configuration file. It is **required** when entered by the operator.

Responses

Incoming service access deleted from network <network_name>.

--WARNING-- Incoming service access deleted from network <network_name>.
Title <title> not found.
{Line repeats for each title not found}

--ERROR-- A network is not defined.

--ERROR-- Network <network_name> is not defined.

--ERROR-- Title <title> may not appear more than once in the list of specified titles.

--ERROR-- Expecting NAME or STRING, found <xxx>.

Examples

```
delete_outgoing_service_access title='SVL118' network_name=$NET_32

Outgoing service access deleted from network $NET_32.
```

DELETE_RIP_SERVICE (DELRS) NETOU Command

Purpose Deletes the definition of RIP (Routing Information Protocol) for a directly connected IP network. The network is referenced by its IP network number.

Format

DELETE_RIP_SERVICE
IP_NETWORK = list 1..4 of 0..255
REMOVE_ROUTES = boolean

Parameters

IP_NETWORK (IN)

Specifies the IP network number of the directly connected network on which RIP is to be deleted.

REMOVE_ROUTES (RR)

Specifies whether the network routes learned by RIP on the specified IP network should be removed from the routing tables. The default is TRUE.

Responses RIP service for network <ip_network> is deleted.

--ERROR-- RIP network <ip_network> is not defined.

Examples

```
delete_rip_service ip_network = (128,5)

RIP service for network 128.5 is deleted.
```

DELETE_SNMP_COMMUNITY (DELSC) NETOU Command

Purpose Deletes support of an SNMP community.

Format

DELETE_SNMP_COMMUNITY
COMMUNITY = string 1..32
IP_ADDRESS = list 1..4 of 0..255

Parameters

COMMUNITY (C)

Specifies the community name for which support is to be deleted.

IP_ADDRESS (IA)

Specifies an IP address associated with the input community name. If this parameter is not specified, incoming PDUs with the specified community name are denied access. If the parameter is specified, only the entry related to the specific IP_ADDRESS/COMMUNITY pair is deleted.

Responses

The SNMP community '<community>' has been deleted.

IP address <ipaddress> for SNMP community '<community>' has been deleted.

--ERROR-- The SNMP Agent is not defined.

--ERROR-- The SNMP community '<community>' is not defined.

--ERROR-- IP address <ip address> for SNMP community '<community>' is not defined.

Examples

```
delete_snmp_community community='manager 1'

  The SNMP community 'manager 1' has been deleted.

delete_snmp_community community='manager 2' ip_address=(125.3.12.42)

  IP address 125.3.12.42 for SNMP community 'manager 2' has been deleted.

delete_snmp_community community='test'

  --ERROR-- The SNMP Agent is not defined.

delete_snmp_community community='community name not defined'

  --ERROR-- The SNMP community 'community name not defined' is not defined.

delete_snmp_community community='manager 3' ip_address=(125.3.12.42)

  --ERROR-- IP address 125.3.12.42 for SNMP community 'manager 3' is not defined.
```

DELETE_TCPIP_ACCESS (DELTA)
NETOU Command

Purpose Deletes TCP/IP access services to a NOS/VE host. The host is referenced by the interface name of the channel interface previously defined for that host. Deleting the TCP/IP access prevents new connections for TCP/IP applications on the NOS/VE host from being made. Existing connections continue processing until terminated.

Format

```
DELETE_TCPIP_ACCESS
   INTERFACE_NAME = name
```

Parameters

INTERFACE_NAME (IN)

Specifies the logical name of the NOS/VE interface used for the TCP/IP access being deleted.

Responses

TCPIP access is deleted for VE interface <interface_name>.

--WARNING-- TCP/IP access is not added for VE interface <interface name>. It cannot be deleted.

--ERROR-- VE interface <interface_name> is not defined.

--ERROR-- Interface name <interface_name> is not a VE channel interface.

Examples

```
delete_tcpip_access interface_name=$mci4

TCPIP access is deleted for VE interface $MCI4.
```

DELETE_TELNET_SERVER (DELTS) NETOU Command

Purpose Deletes the specified TELNET server.

Format

```
DELETE_TELNET_SERVER
   SERVER_NAME = name
```

Parameters

SERVER_NAME (SN)

Specifies the name of the TELNET server to be deleted.

Responses

TELNET server <server_name> is deleted.

--ERROR-- TELNET server <server_name> was not found.

Examples

```
delete_telnet_server server_name=arhnet

TELNET server ARHNET is deleted.
```

DELETE_X25_GW_INCALL (DELXGI)
NETOU Command

Purpose Deletes DTE addresses specified in previously entered ADD_X25_GW_INCALL commands.

Format

DELETE_X25_GW_INCALL
REMOTE_DTE_ADDRESSES = list 1..31 of string 1..15 or OTHER
GATEWAY_NAME = name

Parameters

REMOTE_DTE_ADDRESSES (RDA)

Specifies one or more DTE addresses to be deleted from the allowed DTE addresses. The DTE addresses must have been specified in a previous ADD_X25_GW_INCALL command. The keyword OTHER may be specified.

GATEWAY_NAME (GN)

Specifies the name of the X.25 application-to-application gateway which receives the incoming calls to be validated. The gateway must have been previously defined.

Responses

X.25 incall deleted from gateway <gateway_name>.

--WARNING-- Remote_dte_address <string> does not exist for X.25 gateway <name>.

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <name> is not defined.

--ERROR-- Remote_dte_address can not include <string>. A remote_dte_address may include only digits 0 thru 9.

Examples The following example specifies that DTE address 4082345 is permitted to make incalls only to services on the CDCNET network permitted to 'OTHER' callers, if any.

```
delete_x25_gw_incall, gateway_name  =  X25GW,
remote_dte_address ='4082345'

X.25 incall deleted from gateway X25GW
```

DELETE_X25_GW_OUTCALL (DELXGO) NETOU Command

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 = list 1..255 of name
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

```
delete_x25_gw_outcall t=''PTFS$FOREIGN''

X.25 gateway title PTFS$FOREIGN deleted.
```

DELETE_X25_GW_PVC (DELXGP) NETOU Command

Purpose Deletes the configuration of an X.25 gateway permanent virtual circuit (PVC).

NOTE

If there is an application connection to the PVC, the connection is not deleted until the X.25 Gateway receives a Session disconnect or the peer connection has sent a reset indication with the CAUSE field set to OUT OF ORDER.

Format

DELETE_X25_GW_PVC
TITLE=name
GATEWAY_NAME=name

Parameters

TITLE (T)

Specifies the title that applications can use to access a particular remote application through this gateway. The title is used to support calls from CDCNET systems to remote systems accessed through the X.25 trunk. The PVC associated with this title is deleted.

GATEWAY_NAME (GN)

Specifies the name of the X.25 gateway which provides access to the remote application. The gateway must have been previously defined.

Responses

PVC <pvc> deleted for title <title>, X.25 gateway <gw>.

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway title <title> not found.

Examples

```
delete_x25_gw_pvc title=PVC$FRN gateway_name=x25_gw

PVC <100> deleted for title PVC$FRN, X.25 gateway x25_gw.
```

DELETE_X25_INCALL (DELXI) NETOU Command

Purpose Deletes configured X.25 incall definition entries (added by using the ADD_X25_INCALL command) for one or more trunk names and one or more remote DTE addresses.

Format

DELETE_X25_INCALL
TRUNK_NAME = list 1..32 of name
REMOTE_DTE_ADDRESSES = list 1..31 of string 1..31

Parameters

TRUNK_NAME (TN)

Specifies the set of trunks associated with the X.25 incall definition entries to be deleted. The value ALL is used to delete the specified entries associated with the remote DTE addresses on all trunks previously configured with X.25 incall definition entries. The default is ALL.

REMOTE_DTE_ADDRESSES (RDA)

Specifies the set of remote DTE addresses associated with the X.25 incall definition entries to be deleted. Wild card attributes (as specified by command ADD_X25_INCALL) must exactly match an entry as specified by ADD_X25_INCALL. For example, a value of '123*' matches only on entry '123*'. It does not match all entries with '123' as the first three digits. If a REMOTE_DTE_ADDRESSES value has multiple entries associated with it (due to different values of DTE_ADDRESS_PROTOCOL_ID), all of those entries are deleted. The value OTHER corresponds to any entries defined with the value OTHER. The value ALL is used to delete all X.25 incall definition entries for the trunks specified. The default is ALL.

Responses The response for each trunk is shown separately.

The following response is generated if the value of the remote DTE address parameter specifies a remote DTE address:

> The X.25 incall definition entry for trunk <trunk_name> for the remote DTE address <remote_dte_address> is deleted.

The following response is generated if the value of the remote DTE address parameter specifies a list of two or more remote DTE addresses:

> The X.25 incall definition entries for trunk <trunk_name> for the following remote DTE addresses are deleted.
> {The above line is followed by a list of address prefixes. Please see the command response example for the format in which this information is displayed.}

The following response is generated if a value of ALL is specified for the remote DTE address:

> X.25 incall definition entries for trunk <trunk_name> for all remote DTE addresses are deleted.

The following response is generated if no X.25 incall definition entry has been configured, regardless of input errors:

--WARNING-- No X.25 incall definition entry for any trunk is defined.

The following response is generated if X.25 incall definition entries have been configured, but none were explicitly specified for a trunk:

--WARNING-- No X.25 incall definition entry for trunk <trunk_name> is defined.

Error responses:

--ERROR-- An X.25 incall definition entry for the remote DTE address <remote_dte_address> is not defined for trunk <trunk_name>.

--ERROR-- Trunk name <trunk_name> is not defined.

--ERROR-- Trunk name <trunk_name> is not an X.25 trunk.

--ERROR-- Trunk name <trunk_name> may not appear more than once in the list of specified trunks.

--ERROR-- Remote DTE address <remote_dte_address> may not appear more than once in the list of specified remote DTE addresses.

--ERROR-- The remote DTE address <remote_dte_address> contains invalid characters.

--ERROR-- The remote DTE address <remote_dte_address> has an invalid bracketed expression.

--FATAL-- Not enough memory available for required table space.

Examples

```
delete_x25_incall remote_dte_address=12345

The X.25 incall definition entry for trunk X25_TRUNK
for remote DTE address 12345 is deleted.

delete_x25_incall remote_dte_address=('12345678', '87654321')

The X.25 incall definition entries for trunk X25_TRUNK
for the following remote DTE addresses are deleted.

  12345678
  87654321

delete_x25_incall

X.25 incall definition entries for trunk X25_TRUNK
for all remote DTE addresses are deleted.

delete_x25_incall trunk_name=ACCUNET

No X.25 incall definition entry for trunk ACCUNET is defined.
```

DELETE_X25_SWITCH (DELXS)

Purpose Allows an operator to delete X.25 switch definition entries previously added using the ADD_X25_SWITCH command. Deleting a switch definition does not affect existing connections.

Format

DELETE_X25_SWITCH
GATEWAY_NAME = list 1..32 of name or *ALL*
CALLED_DTE_ADDRESSES = list 1..31 of string 1..31 or *keyword value*

Parameters *GATEWAY_NAME (GN)*

Specifies a set of X.25 Gateways associated with the X.25 switch definition entries to be deleted. Use ALL to delete the specified entries associated with the CALLED_DTE_ADDRESSES on all X.25 Gateways previously configured with X.25 switch definition entries. The default is ALL.

CALLED_DTE_ADDRESSES (CDA)

Specifies one or more called DTE addresses. A DTE address can contain wild card attributes (to associate the switch definition with multiple DTE addresses) as follows:

[D1-D2] Within a bracketed group, a range of digits can be specified. D1 represents the low end of the range; D2 represents the high end. Any value within the range can be used to match a digit in the position specified in the address. For example, 1234[6-7]59 matches two DTE addresses: 1234659 and 1234759.

[...] Within a bracketed group, any list of single digits can be specified. For example, 1234[179]59 matches three DTE addresses: 1234159, 1234759, and 1234959.

* Any single * can be used to match a string of digits, including the null string. For example, 1234* matches 1234, 12341, 1234562345, and so forth.

? Any single ? can be used to match a digit in the position specified in the address. For example, 1234?59 matches 10 DTE addresses: 1234059, 1234159, 1234259 ... 1234959.

If multiple ADD_X25_SWITCH definitions match a given DTE address, the call definition associated with the most specific string is deleted. For example, 123?56 is more specific than 123* in matching called DTE address 123456.

The value OTHER corresponds to any entries defined with the value OTHER. The value ALL deletes all X.25 switch definition entries for the X.25 Gateways specified. The default is ALL.

Responses (The response for each X.25 Gateway is shown separately.)

The following response is generated if the value of the called DTE address parameter specifies a called DTE address:

> The X.25 switch definition entry for X.25 Gateway <gateway_name> for the called DTE address <called_dte_address> is deleted.

The following response is generated if the value of the called DTE address parameter specifies a list of two or more called DTE addresses:

> The X.25 switch definition entries for X.25 Gateway <gateway_name> for the following called DTE addresses are deleted.
>
> <called_dte_address>

The following response is generated if a value of ALL is specified for the called DTE address:

> X.25 switch definition entries for X.25 gateway <gateway_name> for all called DTE addresses are deleted.

The following response is generated if no X.25 switch is defined, regardless of input errors:

> --WARNING-- No X.25 switch definition entry for any X.25 Gateway is defined.

The following response is generated if an X.25 switch is defined, but none explicitly for an X.25 Gateway:

> --WARNING-- No X.25 switch definition entry for X.25 Gateway <gateway_name> is defined.

The following error responses are generated upon an error condition:

> --ERROR-- An X.25 switch definition entry for the called DTE address <called_dte_address> is not defined for X.25 Gateway <gateway_name>.
>
> --ERROR-- X.25 Gateway <gateway_name> is not defined.
>
> --ERROR-- X.25 Gateway <gateway_name> may not appear more than once in the list of specified X.25 Gateways.
>
> --ERROR-- Called DTE address <called_dte_address> may not appear more than once in the list of specified called DTE addresses.
>
> --ERROR-- The called DTE address <called_dte_address> contains invalid characters.
>
> --ERROR-- The called DTE address <called_dte_address> has an invalid bracketed expression.
>
> --FATAL-- Not enough memory available for required table space.

Examples

```
delete_x25_switch called_dte_address=12345
```

The X.25 switch definition entry for X.25 Gateway X25_GW_SWITCH for called DTE address 12345 is deleted.

```
delete_x25_switch called_dte_address=('12345678', '87654321')
```

The X.25 switch definition entries for X.25 Gateway X25_GW_SWITCH for the following remote DTE addresses are deleted.

```
12345678
87654321
```

```
delete_x25_switch
```

X.25 switch definition entries for X.25 Gateway X25_GW_SWITCH for all called DTE addresses are deleted.

```
delete_x25_switch gateway_name=X25_SWITCH
```

No X.25 switch definition entry for X.25 Gateway X25_SWITCH is defined.

DELETE_X25_TERMINAL_GW_OUTCALLS (DELXTGO) NETOU Command

Parameters Deletes the configuration of an X.25 terminal gateway outcall(s).

Format

```
DELETE_X25_TERMINAL_GW_OUTCALLS
   OUTCALL_NAME = name or ALL
   GATEWAY_NAME = name
```

Parameters

OUTCALL_NAME (ON)

Specifies either the outcall name entered on the ADDXTGO command that corresponds to all X.25 outcall block definitions, or ALL.

GATEWAY_NAME (GN)

Specifies the name of the X.25 terminal gateway which provides access to the remote application. The gateway must be previously defined.

Responses

X.25 terminal gateway outcall <outcall_name> is deleted.

X.25 terminal gateway outcalls are deleted.

--ERROR-- X.25 terminal gateway outcall <outcall_name> was not found.

--ERROR-- X.25 terminal gateway <name> is not defined.

Examples

```
delete_x25_terminal_gw_outcalls on=SOLVER1 gn=all

X.25 terminal gateway outcall SOLVER1 is deleted.

delete_x25_terminal_gw_outcalls on=ALL gn=all

X.25 terminal gateway outcalls are deleted.
```

DISPLAY Commands 8

DISPLAY Commands 8

This chapter provides complete descriptions of all CDCNET Network Operations DISPLAY commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

DISPLAY_CHANNEL_CONNECT_STATUS (DISCCS) NETOU Command

Purpose Displays the operating status of the channel connection entity (CCE), the specific service access points (SAPs), and the specific connections serviced by the channel connection entity. Displayed information includes identifying the SAPs and their connections by their SAP names, local reference number and CEPID (Connection End Point Identifier).

Connection identifiers are unique for all connections assigned by CCE. Channel connection entity status displays are described following the command examples.

Format

DISPLAY_CHANNEL_CONNECT_STATUS
DISPLAY_OPTION = keyword value

Parameters *DISPLAY_OPTION (DO)*

Displays the level of status response. The allowed keyword values are:

Keyword Value	Description
SUMMARY (S)	Provides a summary status display of channel connect information per interface.
EXPANDED (E)	Provides an expanded status display of channel connect information per interface.

Default is SUMMARY.

Responses The following header precedes the options display (see examples).

Channel Connection Status.

Within the status display, the following response is inserted if the SAP is open and has no connections established.

No connections established for this SAP.

Examples

```
display_channel_connect_status

Channel Connection Status

interface name = MCI_5
device name = $MCI5
link status = UP
date and time interface last became active = 89/03/13 08.04.01
open SAPs = TAP  SMAP
number of connections = 5
number of connections to host congested = 1
number of connections user initiated stop = 1

interface name = MCI_7
device name = $MCI7
link status = UP
date and time interface last became active = 89/03/13 08.04.01
open SAPs = TAP  SMAP
number of connections = 0
number of connections to host congested = 0
number of connections user initiated stop = 0


display_channel_connect_status display_option=e

Channel Connection Status

interface name = MCI_5

  SAP name = TAP

Local     CEPID     User      Peer/Local  Connection  Connection
Ref #               CEPID       Credits   Priority    State
0001(16)  001068A0  001056A0      4/5     Priority    Active
0002(16)  001069F0  00106890      6/2     Normal      Disconnecting
0003(16)  00106BE0  00107056      3/1     Normal      Congested


  SAP name = SMAP

Local     CEPID     User      Peer/Local  Connection  Connection
Ref #               CEPID     Credits     Priority    State
0005(16)  00105DC0  00000000    16/0      Normal      Con req recvd
0018(16)  001053A0  00106290     3/5      Normal      User stopped


interface name = MCI_7

  SAP name = TAP

No connections established for this SAP

SAP name = SMAP

No connections established for this SAP
```

Channel Connect Display Description

The summary status display information includes:

- Interface name
- Device name
- Link status
- Date and time interface last became active
- Open SAPs
- Number of connections
- Number of connections to host congested
- Number of connections user initiated stop

The expanded connection display information includes:

- Interface name
- For each SAP that is open:

  Local reference number for each connection associated with the given SAP

  CEPID (Connection Endpoint Identifier)

  User CEPID

- Peer allocated credits per connection
- Local allocated credits per connection
- Connection priority
- State of the connections as follows

  Idle - no data sent or received in the last 5 minutes

  Congested - window to host closed

  User stopped - CC user issued a stop on the connection

  Active - normal or expedited data sent/received in the last 5 minutes

  Con req sent - a connection request has been sent for the connection

  Con req recvd - a connection request has been received - awaiting accept or reject from the user

  Disconnecting - the connection is being disconnected

DISPLAY_CLNS_FILTER (DISCF)
NETOU Command

Purpose Displays previously configured CLNS filter entries for a specified network or all networks.

Format **DISPLAY_CLNS_FILTER**
NETWORK_NAME = *name* or *ALL*

Parameters *NETWORK_NAME (NN)*

Identifies the name of a currently defined network within this system or all networks. If a name is specified, CLNS filter entries for the network named are displayed. If the keyword ALL is specified, CLNS filter entries from all networks are displayed. The default is ALL.

Responses All CLNS filter entries for the specified network (or all networks) are displayed. The source and destination address prefixes and the access allowed values are shown for each entry. The display is ordered the same as used during CLNS filter processing (longer address prefixes first).

The following response is generated if no CLNS filter is defined for the specified network, regardless of input errors:

No CLNS filter entries are defined for network <network_name>.

--ERROR-- Network <network_name> is not defined.

Examples

```
display_clns_filter network_name=NETWORK1

 CLNS Filter Entries for network NETWORK1:

  Allowed   Source NSAP Prefix          Destination NSAP Prefix

  FALSE     0800271000EE334567          004700084001BC56
  TRUE      0800271000EE33              004700084001BC56
  TRUE      080027100088                any
  FALSE     any                         004700084003A142000000000000..
                                        ..0056


display_clns_filter network_name=NETWORK1

 No CLNS filter entries are defined for network <network_name>.


display_clns_filter network_name=NETWORK2

 --ERROR-- Network NETWORK2 is not defined.
```

DISPLAY_CLNS_OPTIONS (DISCO) NETOU Command

Purpose Displays the current value of connectionless mode network service (CLNS) options. These values are either the CDCNET default values or the values specified by the CHANGE_CLNS command.

Format **DISPLAY_CLNS_OPTIONS**
DISPLAY_OPTION = list 1..3 of keyword value

Parameters *DISPLAY_OPTION (DO)*

Displays one or more of the CLNS attributes. The allowed keyword values are:

Keyword Value	Description
HEADER_CHECKSUM (HC)	Determines whether the connectionless mode network service (CLNS) entity is to calculate a checksum for the header on an outgoing initial PDU.
DERIVED_PDU_RETENTION_TIME (DPRT)	Displays the time (in seconds) the DI retains any derived PDUs it receives that do not form a complete initial PDU.
PDU_MAXIMUM_LIFETIME (PML)	Displays the maximum number of hops a network PDU can traverse. If a network PDU traverses more than the number of hops specified by the PML parameter, the PDU is discarded and an error PDU is sent back to the originator of the PDU. The error PDU states that the lifetime was exceeded.
ALL (A)	All of the specified keyword values.

Default is ALL.

Responses The following header precedes the options display (see examples).

CLNS options

Examples

```
display_clns_options

CLNS options
header_checksum             = FALSE
derived_pdu_retention_time  = 10
pdu_maximum_lifetime        = 64
```

DISPLAY_COMMAND_INFORMATION (DISCI) NETOU Command

Purpose Displays parameter information of the specified command. The specified command must be one of the available network commands. The information includes parameter names, types, and their default values.

Format

```
DISPLAY_COMMAND_INFORMATION
   COMMAND = name
```

Parameters

COMMAND (C)

Displays the parameters of specified command. 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.

Remarks As seen in the example, the format of the parameter information displayed by the DISCI command may differ from what appears in this manual for the same commands example. The manual text presents the parameter format of the CHANGE_PASSTHROUGH_STATUS command (when specified on the DISCI command) as follows:

INACTIVITY_TIMER = 120..14400 or keyword

The DISCI display usually provides the TYPE information, the permissable range if the parameter is an INTEGER type, and the permissable keyword values of each parameter.

Examples

```
display_command_information command=change_passthrough_status

FROM DI_100085_SL                              34233
inactivity_timer, it:integer 120..14400 or key infinite = infinite
```

DISPLAY_COMMAND_LIST (DISCL)
NETOU Command

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**

Parameters None.

Responses Alphabetical list of network commands (see example).

Examples

```
display_command_list

add_arp_reported_nets                    add_egp_reported_nets
        .                                        .
        .                                        .
        .
unload_module                            write_terminal_message
```

DISPLAY_COMMAND _LIST_ENTRY (DISCLE) NETOU Command

Purpose The DISPLAY_COMMAND_LIST_ENTRY command is an alias of the DISPLAY_COMMAND_LIST command. Like DISPLAY_COMMAND_LIST, this command displays an alphabetical list of network commands.

Format **DISPLAY_COMMAND _LIST_ENTRY**

Parameters None.

Responses Alphabetical list of network commands (see example).

Examples

```
display_command_list_entry

 add_arp_reported_nets                  add_egp_reported_nets
        .                                      .
        .                                      .
        .
unload_module                          write_terminal_message
```

DISPLAY_CONS_INCALL (DISCIC) NETOU Command

Purpose Displays previously configured CONS incall definition entries for one or more NSAP address prefixes.

Format

DISPLAY_CONS_INCALL
ADDRESS_PREFIX = list 1..10 of string 2..40

Parameters

ADDRESS_PREFIX (AP)

Specifies the set of NSAP address prefixes associated with the CONS incall definition entries to display. The elements of each string defining an address prefix must be hexadecimal digits, unless you specify ALL. Use the value ALL to display all previously configured incall definition entries. The default is ALL.

Responses For each requested address prefix for which a CONS incall definition entry has been configured, the configurable parameters are displayed. All entries with the same parameter settings are grouped together. For the format of this display, see the example.

The following response is generated if no CONS incall definition entry has been configured, regardless of input errors:

No CONS incall definition entry for any NSAP address is defined.

The following response is generated if a CONS incall definition entry has not been configured for the requested address prefix:

--ERROR-- A CONS incall definition entry for the NSAP address prefix <address_prefix> is not defined.

The following response is generated if an ADDRESS_PREFIX parameter contains a nonhexadecimal character:

--ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

The following response is generated if an ADDRESS_PREFIX parameter contains an odd number of digits:

--ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

Responses The abbreviation for this command does not follow the convention.

Examples

```
display_cons_incall

CONS Incall Definition Entries:

 NSAP address prefix          08002710O0EE334567
                              004700084001BC56
                              082345F10A
 Accept PDN charges           FALSE

 NSAP address prefix          394455
                              004700084001BC5688
 Accept PDN charges           TRUE

display_cons_incall address_prefix='00AGEF'

--ERROR-- The NSAP address prefix 00AGEF contains non-hexadecimal characters.

display_cons_incall address_prefix='00AC0F'

--ERROR-- A CONS incall definition entry for the NSAP address prefix 00AC0F is not defined.
```

DISPLAY_CONS_OUTCALL (DISCOC) NETOU Command

Purpose Displays previously configured CONS outcall definition entries for one or more NSAP address prefixes.

Format

DISPLAY_CONS_OUTCALL
ADDRESS_PREFIX = list 1..10 of string 2..40

Parameters *ADDRESS_PREFIX (AP)*

Specifies the set of NSAP address prefixes associated with the CONS outcall definition entries to display. The elements of each string defining an address prefix must be hexadecimal digits, unless you specify ALL. Use the value ALL to display all previously configured outcall definition entries. The default is ALL.

Responses For each requested address prefix for which a CONS outcall definition entry has been configured, the configurable parameters are displayed. All entries with the same parameter settings are grouped together. For the format of this display, see the example.

The following response is generated if no CONS outcall definition entry has been configured, regardless of input errors:

> No CONS outcall definition entry for any NSAP address is defined.

The following response is generated if a CONS outcall definition entry has not been configured for the requested address prefix:

> --ERROR-- A CONS outcall definition entry for the NSAP address prefix <address_prefix> is not defined.

The following response is generated if an ADDRESS_PREFIX parameter contains a nonhexadecimal character:

> --ERROR-- The NSAP address prefix <address_prefix> contains non-hexadecimal characters.

The following response is generated if an ADDRESS_PREFIX parameter contains an odd number of digits:

> --ERROR-- The NSAP address prefix <address_prefix> contains an odd number of digits.

Responses The abbreviation for this command does not follow the convention.

Examples

```
display_cons_outcall

CONS Outcall Definition Entries:

 NSAP address prefix          0800271000EE334567
                              004700084001BC56
                              082345F10A
 Remote DTE address           None specified
 Local DTE address            0123456789012
 Add reverse charging         TRUE
 Add fast select              FALSE
 Interface standard           SAME_AS_INTERFACE
 X.25 facilities              None specified
 Local non-X.25 facilities    None specified
 Remote non-X.25 facilities   None specified
 CCITT DTE facilities         None specified
 User Data                    CONS user specified

 NSAP address prefix          394455
                              004700084001BC5688
 Remote DTE address           0987654321012
 Local DTE address            None specified
 Add reverse charging         FALSE
 Add fast select              TRUE
 Interface standard           SAME_AS_INTERFACE
 X.25 facilities              None specified
 Local non-X.25 facilities    None specified
 Remote non-X.25 facilities   66778899
 CCITT DTE facilities         None specified
 User Data                    03010100

display_cons_outcall address_prefix='00AGEF'

--ERROR-- The NSAP address prefix 00AGEF contains non-hexadecimal characters.

display_cons_outcall address_prefix='00AC0F'

--ERROR--  CONS outcall definition entry for the NSAP address prefix 00AC0F is not defined.
```

DISPLAY_CONS_1980_X25_MAPPING (DISC1XM) NETOU Command

Purpose Displays one or more previously configured NSAP addresses used to route incoming CONS/TP0 traffic received from a 1980 X.25 PDN. This command displays routing information that was previously added via the ADD_CONS_1980_X25_MAPPING command.

Format **DISPLAY_CONS_1980_X25_MAPPING**

Parameters None.

Responses The following header precedes the routing information.

CONS 1980 X.25 routing information (the following is repeated for each entry):

Destination NSAP address = <destination_nsap_address>
Destination DTE address = <destination_dte_address>
Source NSAP address = <source_nsap_address>
Source DTE address = <source_dte_address>

The following header occurs if no routing information exists.

No CONS 1980 X.25 routing information exists.

Examples

```
display_cons_1980_x25_mapping

 CONS 1980 X.25 routing information:

 Destination NSAP address = 39840f01bb7b0000000000000108002510008401
 Destination DTE address = 1234567
 Source NSAP address = 39840f01bb7b0000000000000108002510007201
 Source DTE address = 789012


display_cons_1980_x25_mapping

 CONS 1980 X.25 routing information:
   1
 Destination NSAP address = 39840f01bb7b0000000000000108002510008401
 Destination DTE address = Not Specified
 Source NSAP address = 39840f01bb7b0000000000000108002510007201
 Source DTE address = Not Specified


disc1xm

 No CONS 1980 X.25 routing information exists.
```

DISPLAY_DATE_AND_TIME (DISDAT) NETOU Command

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 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 is HMS.

Responses The following header precedes the date and time in selected format (see example).

System date and time

Examples

```
display_date_and_time

System date and time
    14/10/86 15:09:24

display_date_and_time df=mdy,tf=ampm

System date and time
    02/20/86 10:36 AM.
```

DISPLAY_DEVICE_OUTCALL_STATUS (DISDOS) NETOU Command

Purpose Displays the current status of the Device Outcall Service.

NOTE

In future releases, this command will no longer be supported. You should start using the DISPLAY_OUTCALL_GATEWAY_STATUS command as soon as possible.

Format **DISPLAY_DEVICE_OUTCALL_STATUS**

Parameters None.

Responses Device Outcall Service defined and started.

Device Outcall Service not defined.

Examples

```
display_device_outcall_status

Device Outcall Service defined and started.
```

DISPLAY_DIRECTORY_STATUS (DISDS) NETOU Command

Purpose Displays the operating status of the Directory Management Entity (ME) in a DI. The command supports summary, expanded, and detail displays.

Format

DISPLAY_DIRECTORY_STATUS
DISPLAY_OPTION = list 1..3 of keyword value
TITLE = list 1..15 of string or *ALL*

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)	Selects the expanded display. See the status display description following the examples for more information.
DETAIL (D)	Selects the detail display. See 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 detail display with the DISPLAY_OPTION parameter. If you select a summary display, this parameter is ignored. Default is ALL.

Responses The following header precedes the directory status display (see example).

Directory Status

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, see the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
display_directory_status

Directory Status

current registered titles = 4
current received titles = 25
current translation requests = 8

display_directory_status do=e t=''$I_LOG_ME_LOG_GROUP_1''..

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  address = 00001ACB08002510009301BC
  priority = 3
  registered by: TDI_AVCD                        85/11/14  02:15:32

display_directory_status do=e t=''TIMESHARING'',mdi1

Directory Status

title = TIMESHARING
  address = 00001ACB08002510009301BC
  priority = 3
  registered by: CYBER 180 SN312, MODEL82     85/11/14  02:15:32

display_directory_status do=e t=''$I_LOG_ME_LOG_GROUP_2'',..
system=mdi1

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  address = 00001ACB08002510009301BC
  priority = 3
  registered by: $DI_080025011312             85/11/14  02:15:32

display_directory_status do=d t=''$I_LOG_ME_LOG_GROUP_1'',..
system=mdi1

Directory Status

title = $I_LOG_ME_LOG_GROUP_1
  user_info =
  address = 00001ACB08002510009301BC
  priority = 3
  service = GENERIC_TRANSPORT/OSI TRANSPORT
  translation_domain = CATENET
  class = INTERNAL
  dirid = 00001ACB08002510009385111402153219 20
```

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.
`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: CDC_DEFINED_XXX CUSTOMER_DEFINED_XXX OSI_CLNS OSI_SESSION OSI_NON_CDNA_SESSION OSI_PRESENTATION OSI_NON_CDNA_PRESENTATION OSI_TRANSPORT SESSION UNKNOWN VIRTUAL_TERMINAL
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) NETOU Command

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 The following header precedes the DI system status display (see example).

DI System Status.

Remarks A DI containing an MPB-II board cannot contain a PMM board. Instead, the MPB-II RAM is treated as if it were PMM. The DISDSS display gives information on the PMM status, but the memory reflected is actually MPB-II.

Examples In this example, the DISPLAY_DI_SYSTEM_STATUS command is entered, omitting the DISPLAY_OPTION parameter. The command returns a summary status response.

```
display_di_system_status

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
Largest SMM memory extent available = 71602
```

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, see 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	Number of tasks that can run on this system.
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. See 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. See 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 last 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 extents into which available memory is divided.
	Deloadable Memory	The number of bytes that can be used when a deloadable threshold is reached.

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) described later in this chapter.

DISPLAY_EGP_PEER_STATUS (DISEPS) NETOU Command

Purpose Displays the current status of the Exterior Gateway Protocol (EGP) peers and the configured parameters for a specific device interface.

Format

DISPLAY_EGP_PEER_STATUS
DISPLAY_OPTION = keyword value

Parameters *DISPLAY_OPTION (DO)*

Displays the level of status information to be reported for one or more EGP peers. The following keyword values are allowed.

Keyword Value	Description
PEERS (P)	Displays a list of the defined EGP gateway peers and the current status of each gateway.
REPORTED_NETS (RN)	Displays a list of the local network addresses that are reportable to the defined peers.
FIXED_PARAMETERS (FP)	Displays the configured parameters for each of the defined EGP gateway peers.
ALL (A)	Displays all of the specified keyword values.

Default is PEERS.

Responses The following header precedes the requested peer status information (see example).

```
EGP PEERS:
```

If no EGP peers are defined, the following response replaces the display.

```
No EGP peers defined for this DI.
```

The following header precedes the reportable nets display (see example).

```
EGP REPORTABLE NETS:
```

If no reportable nets are defined, the following response replaces the display.

```
No EGP Reportable Nets defined for this DI.
```

The following header precedes the fixed parameter display (see example).

```
EGP PEER'S FIXED PARAMETERS:
```

If no EGP peers are defined, the following response replaces the display.

```
No EGP peers are defined for this DI.
```

Examples

```
Senc c='display_egp_peer_status do=all'

EGP Status

EGP PEERS:
Peer's IP Addr   State    R     S     T1    T2  M
 10.61.00.02       UP    20    300    25   105  P
    :
EGP REPORTABLE NETS:
128.05.00.02
    :
EGP PEER'S FIXED PARAMETERS:
Peer's IP Addr          USE
 10.61.00.02            AUTO

 asn   mhi  mpi  rci  hudt hact anut andt pnut pndt
  78   30   120  30   3600  120  3    1    1    0

 tha  tpa   rha  rpa
  10   10    5    15
    :
```

EGP Peer Status Display Description

The DISPLAY_EGP_PEER_STATUS command supports display status for the DISPLAY_OPTION values of PEERS, REPORTED_NETS, and FIXED_PARAMETERS.

The display option PEERS returns the following information:

Column	Contents
Peer's IP addr	The network address of a directly connected gateway peer that was defined by a DEFINE_EGP_PEER command.
State	The EGP protocol state for the PEER. The following are possible values:

Value	Description
STOP	The peer protocol has not been started.
IDLE	The peer has been started, but no protocol activity is in progress.
AQSN	The gateway periodically retransmits REQUEST commands to the peer to "acquire" the peer.
DOWN	The gateway has received a REQUEST or a CONFIRM response from the peer but the peer is still declared down. The gateway periodically retransmits HELLO commands.
UP	The protocol has declared the peer up. The gateway periodically retransmits HELLO and POLL commands and processes all commands and responses received.
CEASE	STOP_EGP_PEER command causes the gateway to transmit a CEASE command and go to the CEASE state.

R	The receive sequence number.
S	The send sequence number.
T1	The interval between HELLO retransmissions.
T2	The interval between POLL retransmissions.
M	The mode in which the protocol is operating. The possible values are:

Value	Description
A	Active mode
P	Passive mode
?	Mode is unknown at this time (state is STOP, IDLE, AQSN, or CEASE).

The display option REPORTED_NET returns the following values for each reportable network.

Value	Meaning
IP addr	The network address to be reported to all peers.

The display option FIXED_PARAMETER returns the following values for each peer.

Column	Contents
Peer's IP addr	The network address of the peer configured by the DEFINE_EGP_PEER command.
Use	The type of peer that was configured.

Value	Description
AUTO	The peer may be replaced by a BACKUP peer if acquisition is not possible.
MANUAL	The peer may not be replaced by a BACKUP peer if acquisition is not possible.
BACKUP	The peer may be acquired to replace an AUTO peer that cannot be acquired.

asn	Autonomous system number of the local network assigned by SRI International.
mhi	Minimum hello interval in seconds.
mpi	Minimum poll interval in seconds.
rci	Request/cease interval. Time in seconds between retransmission of request or cease commands.
hudt	Hold up/down time-out. Time in seconds the protocol waits for a peer response in the UP or DOWN states.
hact	Hold acquisition/cease time-out. Time in seconds the protocol waits for a peer response in the acquisition or cease states.
anut	Active neighbor up threshold. The threshold used in the neighbor reachability algorithm for an active peer.
andt	Active neighbor down threshold. The threshold used in the neighbor reachability algorithm for an active peer.
pnut	Passive neighbor up threshold. The threshold used in the neighbor reachability algorithm for a passive peer.
pndt	Passive neighbor down threshold. The threshold used in the neighbor reachability algorithm for a passive peer.

Column	Contents
tha	Transmit hello adjust. The number added to the hello transmission interval in seconds.
tpa	Transmit poll adjust. The number added to the poll transmission interval in seconds.
rha	Receive hello adjust. The number subtracted from the minimum receive hello interval in seconds.
rpa	Receive poll adjust. The number subtracted from the minimum receive poll interval in seconds.

DISPLAY_ESIS_ROUTING_OPTIONS (DISERO) NETOU Command

Purpose Returns the current value of OSI ES-IS routing support attributes.

Format

DISPLAY_ESIS_ROUTING_OPTIONS
DISPLAY_OPTION=list 1..7 of key

Parameters *DISPLAY_OPTION (DO)*

Specifies one or more ES-IS routing option values to be displayed. Default is ALL. The allowed keyword values are:

ROUTING_FUNCTION (RF)
ES_CONFIGURATION_TIMER (ECT)
IS_CONFIGURATION_TIMER (ICT)
ROUTING_CACHE_INACTIVITY_TIMER (RCIT)
ENABLE_CHECKSUM (EC)
CONS_ROUTING_SUPPORT (CRS)
ENABLE_CONFIGURATION_NOTIFY (ECN)
ALL

Responses The following header precedes the options display (see example):

ES-IS Routing Support options

Examples

```
display_esis_routing_options

ES-IS Routing Support options
routing_function = COMBINED
es_configuration_timer = 120
is_configuration_timer = 30
routing_cache_inactivity_timer = 300
enable_checksum = FALSE
cons_routing_support = TRUE
enable_configuration_notify = TRUE
```

DISPLAY_ESIS_ROUTING_STATUS (DISERS) NETOU Command

Purpose Displays the operating status of the ES-IS routing entity. This display includes key ES and IS routing data stores, including the data store for the list of known intermediate systems, the list of networks in the catenet, the cache for routing based on the NSAP address, and the list of known end systems.

Format

DISPLAY_ESIS_ROUTING_STATUS
DISPLAY_OPTION=keyword value

Parameters *DISPLAY_OPTION (DO)*

Specifies the level of status for the general status display. The allowed keyword values are:

SUMMARY (S)
NDS (N)
CDS (C)
ISHDS (I)
ESHDS (ES)
EXPANDED (E)

Default is SUMMARY.

Responses The following header precedes the status display (see example):

ES-IS Routing Status

Within the status display, the following response appears if the given data store is empty:

No entries present for this Data Store

Examples

```
display_esis_routing_status

   ES-IS Routing Status

   Network Entity Title = 39840F01BB7B000000000031910800253004200O

   Data Store  Entries  Date/time and reason for last update

   Network     4        90/05/29  07.02.56    Internal IS-IS change
   Cache       3        90/05/29  08.24.03    ISH used 0800253000C1
   ISH         3        90/05/28  20.09.37    ISH PDU 080025300431
   ESH         2        90/05/28  14.55.29    Timeout 080025100083

display_esis_routing_status display_option=e

   ES-IS Routing Status

   Network Entity Title = 39840F01BB7B000000000031910800253004200O

   Data Store  Entries  Date/time and reason for last update

   Network     4        90/05/29  07.02.56    Internal IS-IS change
   Cache       3        90/05/29  08.24.03    ISH used 0800253000C1
   ISH         3        90/05/28  20.09.37    ISH PDU 080025300431
   ESH         2        90/05/28  14.55.29    Timeout 080025100083


   ES-IS Routing Network Data Store ( NDS )

   Network       Cost

   104(16)       0
   1056(16)      0
   3191(16)      10
   3603(16)      3050
```

```
ES-IS Routing Cache Data Store ( CDS )

NSAP Address                                Type        Count    SNPA

47840202000100030055 2AB68802               ESH PDU     257      080060284211
39840F01BB7B000000000031910800253000C101    ISH used    3290     080025300688
39840F01BB7B000000000031910800253004310 1   RD for IS   3290     080025300431


ES-IS Routing ISH Data Store ( ISHDS )

Network Entity Title                        HT          Network  SNPA

39840F01BB7B000000000031910800253000C100    240         3191(16) 0800253000C1
47840202000100030004216688                  1200        200(16)  080060284210
39840F01BB7B000000000031910800253004310 0   480         3191(16) 080025300431


ES-IS Routing ESH Data Store ( ESHDS )

NSAP Address                                HT          Network  SNPA

39840F01BB7B000000000031910800251000720 0   240         3191(16) 080025100072
47840202000100030055 2AB68802               600         200(16)  080060284211
```

DISPLAY_ETHER_NET_OPTIONS (DISENO) NETOU Command

Purpose Displays the configuration of an Ethernet network solution. The network is identified by logical name. If no name is specified, the configuration of all Ethernet networks defined for the system is displayed.

Format

DISPLAY_ETHER_NET_OPTIONS
NETWORK_NAME = list 1..15 of name
DISPLAY_OPTION = list 1..10 of keyword value

Parameters

NETWORK_NAME (NN)

Displays the logical name of an Ethernet network assigned by a DEFINE_ETHER_NET command. Default displays all defined Ethernet networks.

DISPLAY_OPTION (DO)

Specifies one or more of the network attributes for display. The allowed keyword values are:

TRUNK_NAME (TN)
NETWORK_ID (NI)
NETWORK_NAME (NN)
COST (C)
RELAY_ALLOWED (RA)
MULTICAST_NETWORK (MN)
ROUTING_INFO_NETWORK (RIN)
OUTPUT_QUEUE_LIMIT (OQL)
INCOMING_SERVICE_ACCESS (ISA)
OUTGOING_SERVICE_ACCESS (OSA)
ALL (A)

Default is ALL.

Responses The following header precedes the Ethernet network options display (see example).

Ethernet Network options

Within the options display, one of the following responses is inserted if a NETWORK_NAME is not defined, is not an Ethernet network, or no Ethernet networks are defined, respectively.

Network <name> is not defined.

Network <name> is not an ETHERNET network.

No ETHERNET networks are defined for this system.

Examples

```
display_ether_net_options nn=ethernet_2

Ethernet Network options
trunk_name = $ESCI3
network_id = 123456(16)
network_name = ETHERNET_2
cost = 20
relay_allowed = yes
multicast_network = yes
routing_info_network = yes
output_queue_limit = 30000
incoming_service_access = batch_support, file_transfer,
  interactive, network_management
incoming_service_access titles = none
outgoing_service_access = file_transfer, interactive
outgoing_service_access titles = none
```

DISPLAY_ETHER_TRUNK_OPTIONS (DISETO) NETOU Command

Purpose Displays the configuration of Ethernet trunks. An Ethernet trunk is identified by its trunk name. If a trunk name is not entered, the configuration of all Ethernet trunks defined for the system is displayed.

Format

DISPLAY_ETHER_TRUNK_OPTIONS
TRUNK_NAME = list 1..15 of name
DISPLAY_OPTION = list of keyword value

Parameters *TRUNK_NAME (TN)*

Specifies the logical name assigned to the trunk by a DEFINE_ETHERNET_TRUNK command. Default displays all defined Ethernet trunks.

DISPLAY_OPTION (DO)

Specifies one or more of the trunk attributes for display. The allowed keyword values are:

SLOT (S)
TRUNK_NAME (TN)
MAX_FRAME_SIZE (MFS)
INTERFRAME_SPACING (IS)
ALL (A)

Responses The following header precedes the options display (see example).

ETHERNET Trunk options

Within the options display, the following responses are inserted if a trunk_name is not defined, is not an Ethernet trunk, or if no Ethernet trunks are defined.

Trunk <name> is not defined.

Trunk <name> is not an ETHERNET trunk.

No ETHERNET trunks are defined for this system.

Examples

```
display_ether_trunk_options tn=ethernet_cim02

ETHERNET Trunk options
slot = 4
trunk_name = ETHERNET_CIM02
max_frame_size = 1500
interframe_spacing = 96
```

DISPLAY_FILE_LOAD_STATUS (DISFLS) NETOU Command

Purpose Displays the files that have been loaded into DI memory by the LOAD_FILE command. The DISFLS command has no parameters.

The file load status display is described following the command examples.

Format **DISPLAY_FILE_LOAD_STATUS**

Parameters None.

Responses The following header precedes the file load status display (see example).

display_file_load_status

If no files are loaded, you receive the following response.

No files are currently DI memory-resident.

Examples

```
display_file_load_status

File = LVLTUP
   type = USER_PROCEDURE       size =   900 bytes, file opens =     0
   loaded from : CYBER 180 SN312 ,  MODEL19        08/11/88  02:15:32

File = IPGTUP
   type = USER_PROCEDURE       size =  1210 bytes, file opens =     2
   loaded from : CYBER 180 SN312 ,  MODEL19        08/11/88  11:15:31

There are 2110 bytes in 10 buffers in the 2 files displayed.
```

File Load Status Display Description

The file load status display information includes the following information about each file currently resident in DI memory.

File	The name of the file currently in DI memory.
type	The type of file currently in DI memory.
size	The size of the file currently in DI memory.
file opens	The count of the accesses to this file.
loaded from	The name, serial number and system model of the system from which the file was loaded. This information displays only if the file was loaded from a NOS/VE system.

DISPLAY_FILE_SUPPORT (DISFS) (NOS MDI Only) NETOU Command

Purpose Displays 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 is used to support the file access connection. If this parameter is not specified, the default trunk is used. Default is specified by a DEFINE_SYSTEM command.

Responses The following header precedes the DISFS display (see example).

File Types supported for trunk <name>. Connection is <state>.

Within the status display, the following responses 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, see the Systems Programmer's Reference manual, Volume 2, Network Management Entities and Layer Interfaces.

Examples

```
display_file_support

File Types supported for trunk 03.  Connection is active.

USER_PROCEDURE
TERMINAL_PROCEDURE
LOAD_PROCEDURE
LIBRARY
EXCEPTION
DUMP
CONFIGURATION
BOOT
```

DISPLAY_HARDWARE _STATUS (DISHS) NETOU Command

Purpose Displays status of the logic boards in a DI or the status of an ICA-II in a CDCNET host. If you issue the command, with no parameters, to a DI, you receive the status of all boards. If you issue the command, with parameters, to a DI, you receive the status of the specified devices. If you issue the command, with no parameters, to a CDCNET host, you receive the status of the ICA and all the associated interfaces. If you issue the command, with parameters, to a CDCNET host, you receive the status of the specified ICA-II interface.

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 for a DI), or the status of boards specified by device name with the command. If no device name is entered for an ICA-II, the status of the ICA-II board and the ICI/IEI interfaces are displayed. 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 1..30 of name or *ALL*
DISPLAY_OPTION = keyword value

Parameters *DEVICE _NAME (DN)*

The physical name of the device for which status is to be returned. This parameter is optional and has no default value.

DISPLAY_OPTION (DO)

Displays 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 The following header precedes the hardware status display (see example).

Hardware Status

Within the status display, the following response are 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.

```
display_hardware_status

Hardware Status
                                                                    boot ROM
device name  state   status          version  lim/bank/port  type  enab level
$MPB0        on      active          0000(16)                      n/a  160A
$PMM1        on      active          0008(16)                      n/a  160A
$SMM2        on      active          0001(16)  2                   n/a  0000
$MCI3        on      protocol mism.  0000(16)                      yes  050C
$CIM4        on      configured      0001(16)  0,1,2,3             no   2702
$CIM5        down    not config.     0000(16)                      yes  2702
$ESCI6       on      active          0000(16)                      no   0806
$MCI7        on      active          0000(16)                      yes  050C

$LIM0        on      configured      0008(16)  4           RS232
$LIM1        down    configured      0009(16)  4           RS232
$LIM2        on      not config.     0000(16)  2           RS449
$LIM3        on      not config.     0000(16)  2           RS449
$URI4        on      configured
```

This example shows the summary status display for a LIM.

```
display_hardware_status dn=$lim4_port0

Hardware Status
                                                                    boot  ROM
device name  state   status      version   lim/bank/port   type    enab  level
$LIM4_PORT0  down    configured                            ASYNC
```

This example shows the summary status display for a MPB.

```
dishs dn=$mpb0


Hardware Status
                                                                    boot  ROM
device name  state    status     version   lim/bank/port   type    enab  level
$MPB0        on       configured 0000(16)                        n/a   100A
```

This example shows the expanded status display for all LIMs.

```
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        configured                               ASYNC
$LIM0_PORT1     on        configured                               ASYNC
$LIM0_PORT2     on        configured                               ASYNC
$LIM0_PORT3     on        configured                               ASYNC
$LIM1_PORT0     on        configured                               ASYNC
$LIM1_PORT1     on        configured                               ASYNC
$LIM1_PORT2     on        configured                               ASYNC
$LIM1_PORT3     on        configured                               ASYNC
$LIM2_PORT0     on        configured                               ASYNC
$LIM2_PORT1     on        configured                               ASYNC
$LIM2_PORT2     on        configured                               ASYNC
$LIM2_PORT3     on        configured                               ASYNC
$SMM2_BANK0     on        configured              2
$SMM2_BANK1     on        configured              2
```

This example shows the summary status display for an ICA-II system.

```
display_hardware_status

Hardware Status

                                                                  boot  ROM
device name  state    status      version   lim/bank/port   type  enab  level
$ICA         on       active      1300(16)                        n/a   5204
$ICA_ICI     on       configured                          ICI
$ICA_IEI     down     configured                          IEI
```

This example show the summary status display for a specific ICA-II system (DEVICE_NAME=$ICA)

```
display_hardware_status dn=$ica

Hardware Status

                                                                  boot  ROM
device name     state    status      version   lim/bank/port   type  enab  level
$ICA            on       active      1300(16)                        n/a   5204
```

This example shows the summary status display for a specific ICA-II interface (DEVICE_NAME=$ICA_ICI).

```
display_hardware_status dn=$ica_ici

Hardware Status

                                                                  boot  ROM
device name     state    status      version   lim/bank/port   type  enab  level
$ICA_ICI        on       configured                          ICI
```

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, $CIM4 or $ICA. An empty board slot for a major board is assigned the slot number.	
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 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 peripheral processor. 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 a SMM board.

port — The number of ports defined for a LIM board.

type — For LIMs, this field describes 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.

boot enabled — Specifies whether boot is allowed over this device. (Applicable to only ESCI, CIM, and MCI boards.)

ROM level — Specifies the level of the major boards.

DISPLAY_HDLC_NET_OPTIONS (DISHNO) NETOU Command

Purpose Displays the configuration of an HDLC network. You can display the configuration of a single HDLC network or all 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 1..15 of name
DISPLAY_OPTION = list 1..9 of keyword value

Parameters *NETWORK_NAME (NN)*

Logical name of an HDLC network assigned by a DEFINE_HDLC_NET command as described in this manual.

DISPLAY_OPTION (DO)

Selects one or more network attributes for display. The following display options, described in the DEFINE_HDLC_NET command, are allowed.

TRUNK_NAME (TN)
NETWORK_ID (NI)
NETWORK_NAME (NN)
COST (C)
RELAY_ALLOWED (RA)
ROUTING_INFO_NETWORK (RIN)
OUTPUT_QUEUE_LIMIT (OQL)
OUTGOING_SERVICE_ACCESS (OSA)
INCOMING_SERVICE_ACCESS (ISA)
ALL (A)

Default is ALL.

Responses The following header precedes the HDLC network options (see example).

HDLC Network options

Within the options display, one of the following responses is inserted if NETWORK_NAME is not defined, is not an HDLC network, or no HDLC networks are defined, respectively.

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.

```
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
incoming_service_access = file_transfer
incoming_service_access titles = ARH907
outgoing_service_access = file_transfer, interactive
outgoing_service_access titles = none
```

DISPLAY_HDLC_TRUNK_OPTIONS (DISHTO) NETOU Command

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

Parameters *TRUNK_NAME (TN)*

Logical name of an HDLC trunk. Name assigned by a DEFINE_HDLC_TRUNK command as described in this manual.

DISPLAY_OPTION (DO)

Selects one or more trunk attributes for display. The following keywords, described in the DEFINE_HDLC_TRUNK command in this manual, are allowed.

LIM (L)
PORT (P)
LOCAL_ADDRESS (LA)
REMOTE_ADDRESS (RA)
TRUNK_NAME (TN)
OPTIONS (O)
MAX_UNACK_FRAMES (MUF)
SREJ_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)
DRIVER (D)
TRAFFIC_TYPE (TT)
UNSTABLE_TRUNK_DISCONNECT_TIMER (UTDT)
ALL (A)

Default is ALL.

Responses The following header precedes the HDLC trunk option display (see example).

HDLC Trunk options

Within the options display, one of the following responses is inserted if a TRUNK_NAME is not defined, is not an HDLC trunk, or no HDLC trunks are defined, respectively.

Trunk <name> is not defined.

Trunk <name> is not an HDLC trunk.

No HDLC trunks are defined for this system.

Examples

```
display_hdlc_trunk_options tn=HDLC_TRUNK_1

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_unack_frames = 7
srej_queue_size = 4
max_frame_size = 1500
pf_recovery_timer = 500
error_recovery_timer = 2000
retransmission_limit = 20
trunk_speed = 19200
clocking = transmit
interactive_bandwidth = 7
driver = hdlc_normal
traffic_type = CDCNET
unstable_trunk_disconnect_timer = 0
```

DISPLAY_ICA_SYSTEM_STATUS (DISISS) NETOU Command

Purpose Returns general information about the operation of an ICA-II and resource usage in the ICA. The DISISS command is an alias of the DISPLAY_DI_SYSTEM_STATUS command. See the DISDSS command description, earlier in this chapter, for more information.

DISPLAY_IP_FILTER (DISIF)
NETOU Command

Purpose Displays previously configured IP filter entries for a specified IP network or all IP networks.

Format

DISPLAY_IP_FILTER
IP_NETWORK = list 1..4 of 0..255 or *ALL*

Parameters *IP_NETWORK (IN)*

Specifies a particular IP network or all IP networks currently defined within this system. The IP filter entries from this IP network are displayed. If you specify ALL, the IP filter entries from all IP networks are displayed. The default is ALL.

Responses All IP filter entries for the specified IP network or all IP networks are displayed. The IP addresses, IP masks, port numbers, and access allowed values are shown for each entry. The display is ordered the same as for IP filter processing (first in first out).

The following response is generated if no IP filter is defined, regardless of input errors:

No IP filter entries are defined on IP network <ip_network>.

--ERROR-- IP network <ip_network> is not defined.

Examples

```
display_ip_filter ip_network=(129,179,52)

 IP Filter Entries for IP network 129.179.52:

                    Source                                  Destination
Allowed Protocol    IP Address          Port ranges         IP Address          Port ranges
                    Mask                                    Mask

TRUE    any         129.179.000.000     1..126              129,179.63.000      126..255
                    255.255.000.000                         255.255.255.000

FALSE   23..25      any                                     129.179.64.11       1..65535
                                                            255.255.255.255


display_ip_filter ip_network=(129,179,52)

No IP filter entries are defined on IP network 129.179.52.


display_ip_filter in=(129,179,54)

 --ERROR-- IP network 129.179.54 is not defined.
```

DISPLAY_IP_STATUS (DISIS)
NETOU Command

Purpose Displays the current status of the IP routing tables for a DI. The display includes the values configured from the DEFINE_IP_NET and DEFINE_IP_HOST commands. The display includes the routing information from ARP, EGP, and RIP for directly and indirectly connected networks, respectively. The display also indicates if IP Broadcast is enabled for each of the directly connected networks.

IP status displays are described following the command examples.

Format

DISPLAY_IP_STATUS
DISPLAY_OPTION= list of keyword values

Parameters *DISPLAY_OPTION (DO)*

Displays the level of detail the response contains. The parameter allows the following keyword values.

Keyword Value	Description
SUMMARY (S)	Displays the number of networks, hosts, and gateways that are currently in the IP routing tables.
HOSTS (H)	Displays all directly connected host IP addresses and respective hardware addresses (Ethernet or DTE).
NETWORKS (N)	Displays the status of all directly connected IP networks/subnets.
REMOTE (R)	Displays all remote networks/subnets and the associated gateway used to reach it.
EXPANDED (E)	Displays all the information available with the other keywords.

Default is SUMMARY.

Responses The following header precedes the status display (see example).

IP Routing Table Status

If no Routing table is defined (for example, if no locally attached IP networks or hosts are defined), the following response replaces the status display.

No IP Routing table entries are defined for this system.

Examples

```
display_ip_status display_option = expanded

                    IP Routing Table Status

IP          IP          Local   CDC       Remote       Connected
Hosts       Gateways            Hosts     Networks     Networks
5           1           4       1         6            4

Directly Connected Hosts
Type            IP Address        Ethernet/DTE    Owner
                                    Address
IP Host         129.179.1.1       0800253003aa    ARP
IP Host Agent   129.179.1.2       080025300b33    IPSR
CDC Host        129.179.1.51      080025100077    IPSR
Local           129.179.1.50      080025100077    IPSR
Local           129.179.2.50      080025100077    IPSR
IP Host         129.179.2.1       080025100079    ARP
IP Gateway      10.63.0.124       000001246300    IP
Local           10.63.0.1         000001256300    IPSR
IP Host         129.180.0.2       10297021        IPSR
IP Host         129.180.0.3       10297342        IPSR

Directly Connected Networks/Subnets
Network         Type      Status      Subnet   RARP     IP      Trunk
                                      Mask     Status   Brdcst  Name
129.179.1.0     Ethernet  UP          24       ON       ON      $ESCI6
10.0.0.0        X.25      CONGESTED   0        OFF      OFF     DDN_TRUNK
129.179.2.0     HDLC      DOWN        24       ON       OFF     $LIM4_PORT0
129.180.0.0     X.25      UP          0        OFF      OFF     PDN_TRUNK


Remote Networks/Subnets
Network         Gateway Address   Hops   Owner
128.3.0.0       129.179.1.52      1      RIP
26.0.0.0        10.63.0.124       1      EGP
129.179.3.0     129.179.1.52      1      RIP
129.179.4.0     129.179.2.3       3      IPSR
129.181.0.0     129.180.0.2       1      IPSR
129.182.0.0     129.180.0.2       2      IPSR
```

IP Status Display Description

The summary line status display information includes:

IP Hosts	The number of IP hosts currently in the IP Routing tables.
IP Gateways	The number of gateways currently in the IP Routing tables for the DI.
Local	The number of IP addresses configured for the local DI supported by the specified DI.
CDC Hosts	The number of IP addresses configured for the Control Data hosts supported by the specified DI.
Remote Networks	The number of remote (indirectly connected) networks currently in the IP Routing tables.
Connected Networks	The number of directly connected networks currently in the IP Routing tables.

The expanded status displays routing information for directly connected IP networks, the remote networks, and associated gateways used to reach each IP network. The directly connected host information includes the following:

Type	Identifies the type of host to which the address refers. The host may be a foreign host (IP Host), the local DI (Local), a Control Data Host supported by this DI (Control Data Host), or the host for which the DI is the ARP agent (IP Host Agent).
IP Address	Identifies the IP address of all hosts on the directly connected IP networks. Only IP gateway addresses with a zero third byte are displayed. LAN extension IP host addresses are not displayed on X.25 IP networks with a transparent gateway.
Ethernet/DTE Address	Identifies the Ethernet or X.25 DTE address corresponding to the IP address. Ethernet addresses are configured or learned by ARP. IP converts the DTE addresses from the IP address.
Owner	Identifies the software component that created the IP routing table entry. If IPSR is the owner, a configuration command created the entry.

The directly connected networks/subnets information includes the following:

Network	Identifies the IP network number currently in the IP routing tables.
Type	Identifies the type of network; for example, Ethernet, X.25, or HDLC.
Status	Reports the status of the network.
RARP Status	Reports the status of the RARP for the network.
IP Brdcst	Reports the status of the IP Broadcast for the network.
Subnet Mask Size	Reports the number of bits in the subnet mask.
Trunk Name	Reports the name of the CDCNET trunk of the directly connected network/subnet.

The remote network displays the following information about remote networks and subnetworks:

Network	Identifies the IP network number currently in the IP Routing tables.
Gateway Address	Identifies the IP address of a gateway (on a directly connected network) through which the remote network can be accessed from this DI. Also identifies the Ethernet address for an ARP learned gateway.
Hops	Reports the number of hops required to reach the remote network from this DI.
Owner	Identifies the software component that created the IP Routing table entry. If IPSR is the owner, a configuration command created the entry.

DISPLAY_ISIS_ROUTING_STATUS (DISIRS) NETOU Command

Purpose Displays the operating status of the IS-IS Routing Entity (the CDCNET version of IS-IS, not the OSI version of IS-IS). This display includes key routing data stores, including the data store for routing both CLNS and CONS PDUs as well as the data store for received IS-IS PDUs.

Format

DISPLAY_ISIS_ROUTING_STATUS
DISPLAY_OPTION=keyword value

Parameters *DISPLAY_OPTION (DO)*

Specifies the level of status for the general status display. The allowed keyword values are:

SUMMARY (S)
LCRDS (L)
RDCNDS (R)
ALIAS (A)
NAPDS (N)
EXPANDED (E)

Default is SUMMARY.

Responses The following header precedes the status display (see example):

IS-IS Routing Status

Within the status display, the following response appears if IS-IS is configured but the given data store is empty:

No entries present for this Data Store

Examples

```
display_isis_routing_status

IS-IS Routing Status

number of LCRDS entries          = 4
number of RDCNDS entries         = 2
number of inactive networks      = 1
number of congested networks     = 1
number of ALIAS entries          = 1
number of NAPDS prefixes         = 3
maximum hop count to any system = 2

last LCRDS update occurred at 90/05/29  07.02.56 due to a change in
system 0800253000C1 because of a congested network
```

```
display_isis_routing_status display_option=e

IS-IS Routing Status

number of LCRDS entries         = 4
number of RDCNDS entries        = 2
number of inactive networks     = 1
number of congested networks    = 1
number of ALIAS entries         = 1
number of NAPDS prefixes        = 3
maximum hop count to any system = 2

last LCRDS update occurred at 90/05/29  07.02.56 due to a change in
system 0800253000C1 because of a congested network


IS-IS Routing Least Cost Routing Data Store ( LCRDS )

 Destination network 104(16) , number of active paths = 1

  Next hop network/system    PDU count   Hops    Total cost

  104(16)/080025300420       0           1       0

 Destination network 1056(16) , number of active paths = 1

  Next hop network/system    PDU count   Hops    Total cost

  1056(16)/080025300420      0           1       0

 Destination network 3191(16) , number of active paths = 1

  Next hop network/system    PDU count   Hops    Total cost

  3191(16)/080025300420      6821        1       10

 Destination network 3603(16) , number of active paths = 1

  Next hop network/system    PDU count   Hops    Total cost

  3191(16)/0800253000C1      1473        2       3060        CONGESTED


IS-IS Routing Received Directly Connected Network Data Store (RDCNDS)

 System 0800253000C1 , Last Received RIDU Sequence Number 5988

    Network            Cost         Status

    3191(16)           10           ACTIVE
    3603(16)           3050         CONGESTED

 System 080025300431 , Last Received RIDU Sequence Number 656117

    Network            Cost         Status
```

```
   3191(16)            10            INACTIVE
   3603(16)            3050          CONGESTED


IS-IS Routing ALIAS Data Store ( ALIAS )

   System              Inactive Network       Alias Network

   080025300431        3191(16)               3603(16)


IS-IS Routing NSAP Address Prefix Data Store ( NAPDS )

   System 0800253000C1 , Next hop network/system 3191(16)/0800253000C1

   NSAP Address Prefixes

   470004FF1125260800EE65432101
   470004FF1125260800EE12345601
   470003CC
```

IS-IS Routing Status Display Description

The SUMMARY display includes the following operating IS-IS status information:

- Number of Least Cost Routing Data Store (LCRDS) entries
- Number of Received Directly Connected Network Data Store (RDCNDS) entries
- Number of inactive networks in the catenet
- Number of congested networks in the catenet
- Number of ALIAS entries
- Number of NSAP Address Prefix Data Store (NAPDS) prefixes
- Maximum hop count to get to any system in the catenet from this system
- When the last LCRDS update took place (YY/MM/DD HH.MM.SS) and the reason for the update

The Least Cost Routing Data Store (LCRDS) display shows the following for each active network in the catenet:

- Destination network ID
- Number of active paths to the given network

- The following routing information for the given network:
  - Next hop network and system to reach the given network
  - Number of PDUs routed to the given network
  - Number of hops to get to the given network
  - Whether any links are congested in the path between this system and the given network

The Received Directly Connected Network Data Store (RDCNDS) display shows the following for each multi-homed system (other than this one) in the catenet:

- System ID of the system
- Sequence number of the last RIDU received from the system
- The following Directly Connected Network information for the system:
  - Network identifier
  - Network cost
  - Status of the network (INACTIVE, ACTIVE, CONGESTED)
  - If appropriate, an additional attribute (Pseudo Net, Relay Restricted)

The ALIAS display shows the following for each inactive network in the catenet that can be reached by another network:

- System associated with the inactive and alias networks
- Network identifier of the inactive network
- Network identifier of the alias network (the network used for routing PDUs that are destined for the inactive network)

The NSAP Address Prefix Data Store (NAPDS) display shows the following for each known boundary system with NSAP address prefixes in the catenet:

- The boundary system where the given NSAP address prefixes were statically configured
- Next hop network and system to reach the given boundary system
- The list of NSAP address prefixes for the given boundary system (configured by ADD_NON_CDC_ROUTING_ENTRY in that system with DISTRIBUTE=TRUE)

The expanded display is a concatenation of the SUMMARY, LCRDS, RDCNDS, ALIAS, and NAPDS displays.

DISPLAY_LINE_OPTIONS (DISLO)
NETOU Command

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 1..15 of names
DISPLAY_OPTION = list of keyword value

Parameters *LINE_NAME (LN)*

Displays the logical name of one or more lines. 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)

Displays one or more of the line attributes. These attributes are defined by parameters on the DEFINE_LINE command that configured the line. For more information on these parameters, see the DEFINE_LINE command description in this manual. Allowed keyword values include the following.

LIM (L)
PORT (P)
TIP_NAME (TN)
LINE_NAME (LN)
LINE_TYPE (LT)
LINE_SUB_TYPE (LST)
CARRIER_TYPE (CT)
LINE_SPEED (LS)
AUTO_RECOGNITION (AR)
TRANSMISSION_BLOCK_SIZE (TBS)
CONNECTION_CONNECT_TIMEOUT (CCT)
CONNECTION_DISCONNECT_TIMEOUT (CDT)
TERMINAL_DEFINITION_PROCEDURE (TDP)
TERMINAL_USER_PROCEDURE (TUP)
USER_CONNECTION_LIMIT (UCL)
EIA_FLOW_CONTROL (EFC)
CLOCKING (C)
DATA_PARITY (DP)
VALIDATE_USERS (VU)
ALL (A)

Default is ALL.

Responses The following header precedes the options display (see example).

Line options

Within the options display, one of the appropriate following responses appears if the LINE_NAME is not defined or if no lines are defined for the system, respectively.

Line_name <name> is not defined.

No lines are defined for this system.

Examples

```
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 = off
clocking = internal
data_parity = even
validate_users = yes
```

DISPLAY_LINE_STATUS (DISLS) NETOU Command

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 appears twice in the status display.

NOTE

For the X.25 TIP, use DISPLAY_X25_INTERFACE_STATUS to obtain the correct line status.

Line status displays are described following the command examples.

Format

DISPLAY_LINE_STATUS
LINE_NAME = list 1..32 of name
TIP_NAME = list 1..7 of name
LINE_STATE = list 1..5 of keyword value
DISPLAY_OPTION = list 1..3 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.

Keyword Value	Description
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 (A)	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 are displayed. If you specify TIP_NAME, all lines supported by the specified TIP that are in the selected line state are 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 The following header precedes the line status display (see examples).

Line Status

Within the status display, one of the following responses is 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.

```
display_line_status

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.

```
display_line_status ln=compsci_02

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.

```
display_line_status ln=(line01,line10),do=e

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.

```
display_line_status ln=line01,do=d

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_LLC_2_STATUS (DISL2S)
NETOU Command

Purpose Displays the operating status of the Logical Link Control 2 (LLC 2) entity by displaying the status of specific connections serviced by LLC 2.

Format **DISPLAY_LLC_2_STATUS**

Parameters None.

Responses The following header precedes the status display (see example).

LLC 2 Status

Within the status display, the following response is inserted if the SAP is open and has no connections established.

No connections established for this SAP

Examples

```
display_llc_2_status

LLC 2 Status

SAP name = CONS

 Connection    CEPID      User       Peer MAC       V(R)  V(S)  Queue size
 State                    CEPID      Address                    ACK / Transmit

 NORMAL        001798A0   001786A0   080025300674   001   115   016       0001
 AWAIT_REJECT  001539F0   00122890   080025100080   068   033   002       0005
 D_CONN        00166BE0   00141056   080025300335   000   000   000       0000
```

LLC 2 Status Display Description

The DISL2S command displays the following operating status.

connection state	Operational state of the LLC 2 connection which may be:	
	SETUP	A connection request has been sent for the connection.
	NORMAL	A data transfer state (as are the next five states).
	BUSY	The local system is buffer congested.
	REJECT	Out-of-sequence data has been received from the peer.
	AWAIT	One or more timers have expired on this connection.
	AWAIT_REJECT	Combination of states AWAIT and REJECT.
	D_CONN	The connection is being disconnected.
	RESET	The connection is being resynchronized.
	ERROR	The connection is recovering from a locally detected protocol error.
CEPID	Address of the Connection End Point Identifier table.	
User CEPID	Address of the user Connection End Point Identifier table.	
Peer MAC address	The Medium Access Control (MAC) address associated with the peer.	
V(R)	The LLC 2 receive state variable (next sequence number expected).	
V(S)	The LLC 2 send state variable (next sequence number assigned).	
Queue size ACK	The number of messages in the acknowledge queue.	
Queue size transmit	The number of messages in the transmit queue.	

DISPLAY_LLC_2_TRUNK_OPTIONS (DISL2TO) NETOU Command

Purpose Displays the configuration of LLC 2 trunks. An LLC 2 trunk is identified by its trunk name. If a trunk name is not entered, the configuration of all LLC 2 trunks defined for the system is displayed.

Format

DISPLAY_LLC_2_TRUNK_OPTIONS
TRUNK_NAME = list 1..15 of name
DISPLAY_OPTION = list 1..10 of key

Parameters *TRUNK_NAME (TN)*

Specifies the logical name of an LLC 2 trunk assigned by a DEFINE_LLC_2_TRUNK command.

DISPLAY_OPTION (DO)

Specifies one or more of the trunk attributes for display. The default is ALL. The allowed keyword values are:

ETHER_TRUNK_NAME (ETN)
PEER_MAC_ADDRESS (PMA)
PEER_LSAP_ADDRESS (PLA)
TRUNK_NAME (TN)
MODE (M)
RECEIVE_WINDOW_SIZE (RWS)
TRANSMIT_WINDOW_SIZE (TWS)
ACKNOWLEDGEMENT_TIMER (AT)
POLL_BIT_TIMER (PBT)
REJECT_TIMER (RT)
BUSY_STATE_TIMER (BST)
ALL

Responses The following header precedes the status display (see example):

LLC 2 Trunk options

Within the options display, the following responses are inserted if a TRUNK_NAME is not defined or is not an LLC 2 trunk, or if no LLC 2 trunks are defined.

Trunk <name> is not defined.

Trunk <name> is not an X.25 trunk.

No X.25 trunks are defined for this system.

Examples

```
display_llc_2_trunk_options trunk_name=pdn_relay_device

LLC 2 Trunk options

ether_trunk_name = BACKBONE
peer_mac_address = 0800253006B8
peer_lsap_address = FE(16)
trunk_name = PDN_RELAY_DEVICE
mode = RESTART
receive_window_size = 16
transmit_window_size = 16
acknowledgement_timer = 3000
poll_bit_timer = 3000
reject_timer = 5000
busy_state_timer = 10000
```

DISPLAY_LOGICAL_NAMES (DISLN)
NETOU Command

Purpose Displays 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)*

Displays one or more types of logical names for display. The following keyword values are allowed.

TRUNK_NAME (TN)
NETWORK_NAME (NN)
LINE_NAME (LN)
TIP_NAME (TIP)
ALL (A)

Default is ALL.

Responses The following header precedes the system logical names display (see example).

System logical names

Examples

```
display_logical_names

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

Tip_names
   ASYNCTIP
```

DISPLAY_MEMORY (DISM)
NETOU Command

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..65535
BYTE_COUNT = 1..4096
REPEAT_COUNT = 1..4096

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 (zero).

BYTE_COUNT (BC)

Displays the number of bytes in each line. Use with the REPEAT_COUNT parameter (described next) to specify the number of bytes to display. Default number of bytes to display is 1 (one). Default number of bytes per line is 16.

REPEAT_COUNT (RC)

Displays the number of lines selected. 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 match the specified REPEAT_COUNT, only 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 is the number required to display BYTE_COUNT times the REPEAT_COUNT number of bytes at 16 bytes per line.

Responses The following header precedes the display of memory (see example).

Memory displayed

The following responses may appear when a failure occurs.

--WARNING-- Some memory to be displayed is not in a valid address range.
Displayed memory length truncated.

--WARNING-- Bus error encountered, display terminated.

--WARNING-- Too much memory was requested for display.

--ERROR-- First address to be displayed is not in a valid address range.

--ERROR-- Address xxx not found.

Examples

```
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_NAME_RESOLVER_STATUS (DISNRS) NETOU Command

Purpose Displays the current Domain Name Resolver (DNR) status.

Format DISPLAY_NAME_RESOLVER_STATUS

Parameters None.

Responses display_name_resolver_status complete

--ERROR-- Domain Name Resolver is not defined.

Examples

```
display_name_resolver_status

      TCP/IP Domain Name Resolver Status

   Number of requests timed out     = 10
   Total number of requests issued = 100
   Maximum query retry count        = 5
   Timeout value                    = 10
   Current Default Domain           = SVL.CDC.COM
   Name Server Addresses            = 192.12.251.68
                                      192.12.251.35
                                      192.12.251.33
display_name_resolver_status complete.
```

DISPLAY_NET_DELAY_MEASUREMENT (DISNDM) NETOU Command

Purpose Displays the current measurement parameters and the last calculated average delay time for a running measurement.

Format DISPLAY_NET_DELAY_MEASUREMENT
DESTINATION_SYSTEM = list of name or *ALL*

Parameters *DESTINATION_SYSTEM (DS)*

Specifies the system name, which the network delay measurement is to be made, from the source (originating) system. This is the name as designated by the DESTINATION_SYSTEM parameter on the START_NET_DELAY_MEASUREMENT command.

Default is ALL.

Responses The following header precedes the display response (see example).

Network Delay Measures:

The following messages indicate a failure.

--ERROR-- Network Delay Measurement to <destination_system> is not active.

--ERROR-- System <destination_system> is unknown.

--ERROR-- Destination system <destination_system> is not a CDCNET OSI system.

--ERROR-- No Active Measurements Found.

Examples

```
senc 'display_net_delay_measurement destination_system=ndi_d2..
      dtt=450 imi=15' s=mdi_a1

FROM MDI_A1
  Network Delay Measures:

  To NDI_1
    Delay_time_threshold (msec.) = 450
    Average_only = NO
    Measurements = CONTINUOUS
    Messages_per_measurement = 25
    Measurement_priority = INTERACTIVE
    Inter_measurement_interval (minutes) = 15
    Error_threshold = 1
    Message_timeout (seconds) = 120

    Measurements completed = 12
    Last delay time average (msec.) = 47
      at 13:43:22 on 01/16/90
```

DISPLAY_NETWORK_STATUS (DISNS)
NETOU Command

Purpose Displays the status of network solutions connected to a DI. The command returns status for specific network solutions, or, if you do not specify a network name 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. Default is a status display of all network solutions connected to the DI.

Responses The following header precedes the network status display (see examples).

Network Status

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 The following example displays the network status for ESCI.

```
display_network_status

Network Status
network_name = ESCI_NET
network_type = ESCI
network_id = 00000001(16)
network_status = active
network_cost = 000A(16)
trunk_name = ESCI_TRUNK
trunk_status = up
number of trunk transitionsin the last 30 minutes = 0
device_name = $ESCI6
average time network is congested = 0 %
date and time network last became active = 88/03/13 08.04.01
```

The following example displays the network status with an ICA.

```
display_network_status

Network Status
network_name = ICA_IEI_NET
network_type = Ethernet
network_id = 00000001(16)
network_status = active
network_cost = 0064(16)
trunk_name = IEI_TRUNK
trunk_status = up
number of trunk transitions in the last 30 minutes = 0
device_name = $ICA_IEI
average time network is congested = 0 %
date and time network last became active = 88/03/13   08.04.01
```

The following example displays the network status for HDLC.

```
display_network_status

Network Status
network_name = HDLC_NET
network_type = HDLC
network_id = 0000BABA(16)
network_status = active
network_cost = 037D(16)
trunk_name = HDLC_TRUNK_A
trunk_status = up
number of trunk transitions in the last 30 minutes = 0
device_name = $LIM0_PORT0
trunk_name = HDLC_TRUNK_B
trunk_status = up
number of trunk transitions in the last 30 minutes = 1
device_name = $LIM0_PORT1
average time network is congested = 0 %
date and time network last became active = 88/03/13 08.04.01
```

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 but communications are not being carried on. This occurs when the network solution was in use, but is no longer. The operator may have stopped the network, or the other side of an HDLC or X.25 connection stopped communicating.
- `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_xxxx_NET command in this manual 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. Only HDLC networks return this state.

`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.

`trunk_name` The name assigned to the trunk(s) to which the network solution is connected.

`trunk_status` — Operational status of the trunk. Possible values include the following:

`up`	Active communications are being carried over the trunk.
`down`	Something is physically wrong with the trunk. Usually this indicates a hardware fault in the trunk's interface. This also may indicate an ESCI trunk defined on an ESCI card with the Ethernet cable disconnected.

`trunk_status (HDLC only)` — Operational trunk status for HDLC only. Possible values are:

`up for remote load`	The trunk is currently being used to remotely load a DI over the trunk.
`down -- trunk inoperative`	Too many transmission errors are occurring on the physical line associated with the trunk.
`down -- remote disconnect`	The trunk interface has received a disconnect request from the peer at the other end of the trunk.
`down -- protocol timeout`	The HDLC trunk interface at this end of the trunk has not received anything from the peer at the other side of the trunk within the protocol timeout period.
`down -- trunk unstable`	The HDLC trunk has gone into an inoperable state more than 3 times in the past 5 minutes, or 6 times in the last 30 minutes. The trunk has been taken down for the time interval specified on the DEFHT commands UTDT parameter.
`enabled`	The trunk has been idled by DI software. This may occur when a rotary network is used to remote load a DI. Only one trunk from the rotary network is used to do the loading;, the remaining trunks are put into enabled state until the remote loading has completed.

number of trunk transitions in last N minutes	The number of times trunk status has changed from active to inactive or inactive to active in the past N minutes.
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 or ICA. For X.25 networks, the board type is CIM.
average time network is congested	The percent of time the network is congested.
last operational transition	The date and time operational state of network last changed. Displays either the date and time the network last became active, or the date and time the network was last active. Date and time in YY/MM/DD HH.MM.SS format.

DISPLAY_NON_CDC_ADDRESS_PREFIX (DISNCAP) NETOU Command

Purpose Displays one or more previously configured address prefixes of non-CDCNET NSAP addresses and related next hop information.

The command does not display the next hop information obtained from the ES-IS routing protocol.

Format **DISPLAY_NON_CDC_ADDRESS_PREFIX**
ADDRESS_PREFIX = list 1..10 of string 2..40 or *ALL*

Parameters *ADDRESS_PREFIX (AP)*

Displays the address prefix of non-Control Data NSAP addresses. Enter the address prefix in hexadecimal digits, unless you specify the keyword ALL. ALL displays all previously configured address prefixes. The parameter default is ALL.

Responses The following response is displayed if no address prefix has been previously configured or ES-IS routing is not defined for the system.

No non-CDCNET NSAP address prefix is defined.

ES-IS routing is not defined.

The following response is displayed if ES-IS routing is not defined for the system.

ES-IS routing is not defined.

The following response is displayed if a requested address prefix was not previously configured.

The NonCDCNET NSAP address prefix <address_prefix> is not defined.

The following response is displayed for each value of the address_prefix parameter that contains a nonhexadecimal character.

The NonCDCNET NSAP address prefix (address_prefix) contains nonhexadecimal characters.

The following response is displayed for each value of the address-prefix parameter containing an odd number of digits.

The NonCDCNET NSAP address prefix <address-prefix> contains an odd number of digits.

The address prefix and the next hop (if defined) is displayed for each requested address prefix which does not contain nonhexadecimal characters, contains an even number of digits, and has been configured. See the example for the format of this display.

Examples

```
display_non_cdc_address_prefix

      NonCDCNET NSAP ADDRESS PREFIXES

      NonCDCNET NSAP address prefix      Subnet ID  Next Hop SNPA
      044723                             0001       0800271000EE
                                         0001       080027100123
      047724                             0002       0800281000CC
      04470053                           0001       0800271000AA
      044EFAAA                           Not Specified

display_non_cdc_address_prefix address_prefix='00agef'

The NonCDCNET NSAP address prefix 00agef contains
non-hexadecimal characters.


display_non_cdc_address_prefix ap='00ac0f'

The NonCDCNET NSAP address prefix 00ac0f is not
defined.
```

DISPLAY_NON_CDC_ROUTING_ENTRY (DISNCRE) NETOU Command

Purpose

Display previously configured routing entries for one or more non-CDCNET NSAP address prefixes. This command does not display the routing information obtained via the ES-IS routing protocol.

For this release, this command displays only the routing entries explicitly configured via the ADDNCRE command in the system where the display command is being executed.

Format

DISPLAY_NON_CDC_ROUTING_ENTRY
ADDRESS_PREFIX = list 1..10 of string 2..40 or *ALL*

Parameters

ADDRESS_PREFIX (AP)

Specifies the address prefix of non-CDCNET NSAP addresses for the routing entry to be displayed. The elements of each string defining an address prefix must be hexadecimal digits, unless ALL is specified. Default is ALL, which displays all previously configured routing entries.

Responses

The following response is generated if no routing entry has been configured.

> No routing entry for any NonCDCNET NSAP address is defined.

The following response is generated if a routing entry has not been configured for the requested address prefix.

> A routing entry for the NonCDCNET NSAP address prefix <address_prefix> is not defined.

The following response is generated for each value of the address_prefix parameter which contains a non-hexadecimal character.

> The NonCDCNET NSAP address prefix <address_prefix> contains non-hexadecimal characters.

The following response is generated for each value of the address_prefix parameter which contains an odd number of digits.

> The NonCDCNET NSAP address prefix <address_prefix> contains an odd number of digits.

For each requested address prefix which contains an even number of hexadecimal digits and for which a routing entry has been configured, the next hop trunk and SNPA address are displayed (see the example for the format).

Examples

```
display_non_cdc_routing_entry

CDCNET BOUNDARY SYSTEM NON-CDC ROUTING ENTRIES

  NSAP address prefix          0800271000EE334567
                               004700084001BC56
                               082345F10A
  Next hop trunk               Ether_trunk_one
  Next hop SNPA address        082005470001
  Distribute                   TRUE

  NSAP address prefix          0344271000EE33
                               0324AA00CD
  Next hop trunk               Ether_trunk_two
  Next hop SNPA address        0320054700AC
  Distribute                   FALSE

  NSAP address prefix          00390001
  Next hop trunk               X25telenet_one
  Next hop SNPA address        032005470099345
  Distribute                   TRUE
```

```
display_non_cdc_routing_entry ap='00AGEF'

The NonCDCNET NSAP address prefix 00AGEF contains non-hexadecimal characters.
```

```
display_non_cdc_address_prefix ap='00AC0F'

Routing entry for the NonCDCNET NSAP address prefix 00AC0F is not defined.
```

DISPLAY_NP_GW_OPTIONS (DISNGO) NETOU Command

Purpose Displays the configuration of an application-to-application (A-to-A) NP transparent gateway to a NOS host. The display returns the title or titles registered by the gateway in the catenet and the definition required for A-to-A connections to a NOS host.

Format

DISPLAY_NP_GW_OPTIONS
GATEWAY_NAME = list 1..15 of name
DISPLAY_OPTION = list of keyword value

Parameters *GATEWAY_NAME (GN)*

Displays the logical name of one or more NP transparent gateway definitions. If GATEWAY_NAME is not specified, the configuration of all NP transparent gateways defined for the system is displayed.

DISPLAY_OPTION (DO)

Displays one or more of the gateway attributes. These attributes are defined by parameters on the DEFINE_NP_GW command that configured the line. For more information on these parameters, see the DEFINE_NP_GW command description in this manual. The allowed keyword values are:

GATEWAY_NAME (GN)
TRUNK_NAME (TN)
NOS_PROTOCOL_ID (NPI)
CDCNET_PROTOCOL_ID (CPI)
NOS_DHOST_NUMBER (NDN)
TITLE (T)
ALL (A)

Default is ALL.

Responses The following header is followed by the display of options (see example).

Network Products Gateway options

Within the options display, the following response appears if the GATEWAY_NAME was not previously defined.

NP gateway <name> is not defined.

Within the options display, the following response appears if no NP applications are defined in the system.

No NP gateways are defined for this system.

Examples

```
display_np_gw_options

Network Products Gateway options
gateway_name = A_TO_A
trunk_name = 907_CHANNEL_170
nos_protocol_id = (C0(16),C1(16))
cdcnet_protocol_id = (C4(16))
nos_dhost_number = (1A(16))
title = (109_A_TO_A,109_APPLICATION)
```

DISPLAY_NP_INTERFACE_OPTIONS (DISNIO) NETOU Command

Purpose Displays the network products configuration parameters for the block protocol (BIP and SVM) interface to a NOS system.

Format

DISPLAY_NP_INTERFACE_OPTIONS
INTERFACE_NAME = name
DISPLAY_OPTION = list of keyword value

Parameters *INTERFACE_NAME (IN)*

Displays the logical name of the NP interface. If you do not specify an INTERFACE_NAME, the command displays all defined NP interfaces.

DISPLAY_OPTION (DO)

Displays one or more of the interface attributes. These attributes are defined by parameters on the DEFINE_NP_INTERFACE command that configured the line. For more information on these parameters, see the DEFINE_NP_INTERFACE command description in this manual.

TRUNK_NAME (TN)
MDI_NODE (MN)
COUPLER_NODE (CN)
INTERFACE_NAME (IN)
CONGESTED_THRESHOLD (CT)
ALL (A)

Default is ALL.

Responses The following header precedes the display of options (see example).

Network Products Interface options

Within the options display, the following response appears if the INTERFACE_NAME was not previously defined.

NP interface <interface_name> is not defined.

Within the options display, the following response appears if no NP interfaces are defined in the system.

No NP interfaces are defined for this system.

Examples

```
display_np_interface_options

Network Products Interface options
trunk_name = 907_CHANNEL_170
mdi_node = 03(16)
coupler_node = 19(16)
interface_name = C170_COUPLER_1
congested_threshold = 40
```

DISPLAY_OPERATOR_SUPPORT (DISOS) (NOS MDI Only) NETOU Command

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 a specified 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)*

The trunk name of the logical link for which operator support is to be displayed. 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 The following header precedes the DISOS display (see example).

Operators supported for trunk <name>. Connection is <state>.

Within the status display, the following response appears if the operator support was not defined for a specified trunk.

Operator Support is not defined for trunk <name>.

Within the status display, the following response appears if no operator support was defined for the system.

Operator Support is not defined for the system.

Within the status display, the following response appears if no operators are logged in for the specified system.

-None- replaces the list of operator identifiers if no operators are logged in.

Examples

```
display_operator_support

Operators supported for c170_trunk_03.  Connection is active.
oper1
oper2
oper3
 .
 .
opern
```

DISPLAY_OSI_TRANSPORT_OPTIONS (DISOTO) NETOU Command

Purpose Displays the current value of the OSI Transport options. These values are either the CDCNET defaults or the values specified by the CHANGE_OSI_TRANSPORT command.

Format

DISPLAY_OSI_TRANSPORT_OPTIONS
DISPLAY_OPTION = list of keyword value

Parameters *DISPLAY_OPTION (DO)*

Displays one or more of the OSI transport attributes. These attributes are defined by parameters on the CHAOT command. For more information on these parameters, see the CHAOT command description in this manual. The allowed keyword values are:

CHECKSUMMING (C)
ACKNOWLEDGE_TIMER (AT)
MAXIMUM_WINDOW_SIZE (MWS)
MAXIMUM_CONNECTIONS (MC)
ENABLE_OVER_CONS (EOC)
CONS_USER_DATA (CUD)
ALL (A)

Default is ALL.

Responses The following header precedes the options display (see example).

OSI Transport options

Examples

```
display_osi_transport_options

OSI Transport options
Checksumming = FALSE
Acknowledge timer = 5
Maximum window size = 16
Maximum connections = INFINITE
Enable_over_cons = TRUE
Cons_user_data = NONE
```

DISPLAY_OSI_TRANSPORT_STATUS (DISOTS) NETOU Command

Purpose Displays the operating status of various elements of the OSI Transport, including the status of the specific service access points (SAPs) and of the specific connections serviced by the OSI Transport.

SAPs and connections are specified by their Transport Selectors and references (these values are displayed in summary and expanded status displays). If you specify both transport selectors and references, the command returns status for all specified transport selectors and all specified references.

Format

DISPLAY_OSI_TRANSPORT_STATUS
SAP = list 1..15 of 0..0FFFF(16)
REFERENCE = list 1..15 of 1..0FFFF(16)
DISPLAY_OPTION = keyword value

Parameters

SAP (S)

Displays the transport service access point (TSAP) selector of the SAP whose status is to be displayed.

REFERENCE (R)

Displays the identifier of the connection whose status is to be displayed.

DISPLAY_OPTION (DO)

Displays the level of status for the general status. The parameter is ignored if you specify either a SAP identifier or reference. The following values are allowed for the DISPLAY_OPTION.

Keyword Value	Description
SUMMARY (S)	Selects the summary status display.
EXPANDED (E)	Selects more detailed transport status. Provides additional information about specific SAPs.

Default is SUMMARY.

Responses The following header precedes the status display (see example).

OSI Transport Status

Within the status display, you receive the following responses if a SAP identifier or reference is unknown, or if a SAP has no connections established.

SAP <0..0ffff(16)> unknown

Reference <0..0ffff(16)> unknown

No connections established for this SAP

Examples This command returns a summary display of the OSI transport.

```
display_osi_transport_status

OSI Transport Status
number of SAPs = 2
number of class 0 connections = 2
number of class 2 connections = 0
number of class 4 connections = 8
number transport congested = 0
number user congested = 0
number incoming connections = 4
number outgoing connections = 4
```

This command returns an expanded status display of the OSI transport.

```
display_osi_transport_status display_option=e

OSI Transport Status
number of SAPs = 2
number of class 0 connections = 2
number of class 2 connections = 0
number of class 4 connections = 8
number transport congested = 0
number user congested = 0
number incoming connections = 4
number outgoing connections = 4
```

SAP	SAP name	number of connections	transport congested	user congested
1046(16)	IND_FILE_ACCESS	2	0	0
1B3C(16)	IVT_GATEWAY	2	1	0

This command returns the status of the specified TSAP selector.

```
display_osi_transport_status sap=(1046(16),1b3c(16))

OSI Transport Status

SAP 1046(16)

  reference = 47AC(16)                peer reference = 0224(16)
  peer NSAP address = 000000020800252A022313FE
  peer TSAP selector = 0973(16)

  reference = 2391(16)                peer reference = 6363(16)
  peer NSAP address = 000000080800252A0653739E
  peer TSAP selector = 0222(16)

SAP 1B3C(16)
This SAP closed, no new connection may be established with this SAP.

  reference = 8561(16)                peer reference = 6458(16)
  peer NSAP address = 000000080800252A06535473
  peer TSAP selector = 0630(16)
```

This command displays the status of the specified SAP reference identifier for a class 4 connection over CLNS.

```
display_osi_transport_status reference=47ac(16)

OSI Transport Status

reference = 47AC(16), Transport class 4 connection over CLNS
  connection state =  open
  connection table address = 110C28(16)
  user connection identifier = 142042(16)
  Transport selector = 1046(16)
  network service priority = 4
  next assignable transmit sequence number = 00000012(16)
  next expected receive sequence number    = 000002BA(16)
  credits granted to the peer = 3        credits granted from the peer = 16
  average round trip is = 375ms          length data receive queue = 0
  length normal data queue = 0           length normal ack queue = 1
  local acknowledgement timer = 5000ms   delayed acknowledgements = 1
  connection has no expedited data outstanding
  connection is not user congested
  connection is not transport congested
  this is an incoming connection
  Transport PDUs are checksummed
```

This command displays the status of the specified SAP reference identifier for a class 0 connection over CLNS.

```
display_osi_transport_status reference=8561(16)

OSI Transport Status

reference = 8561(16), Transport class 0 connection over CONS
  connection state = open
  connection table address = 120DE0(16)
  user connection identifier = 104064(16)
  Transport selector = 1B3C(16)
  network service priority = 4
  connection is not user congested
  connection is not transport congested
  this is an incoming connection
```

Summary Transport Status Display Description

The summary transport status provides the following information for each OSI transport. This display is the default display.

number of SAPs	The number of SAPs associated with the OSI transport.
number of connections	The number of connections.
number transport congested	The number of connections on which the transport window has closed.
number user congested	The number of connections on which the user requested a stop to message indications.
number incoming connections	The number of connections inbound to the OSI transport.
number outgoing connections	The number of connections outbound from the OSI transport.

Expanded Transport Status Display

The expanded transport status display returns the following information.

number of SAPs	The number of SAPs associated with the OSI transport.
number of connections	The number of connections.
number transport congested	The number of connections on which the transport window has closed.
number user congested	The number of connections on which the user requested a stop to message indications.
number incoming connections	The number of connections inbound to the OSI transport.
number outgoing connections	The number of connections outbound from the OSI transport.
SAP id	The TSAP Selector (SAP identifier) of each SAP serviced by the OSI transport.
SAP name	The name associated with each SAP serviced by the OSI transport.
number of connections	The number of connections in each SAP.
transport congested	The number of connections for which the transport window has closed.
user congested	The number of connections on which the user has requested a stop of message indications.

Transport Selector Status Display

The transport selector status display provides the following information about the specified service access point identifier (SAP ID).

SAP status	If the SAP is closed, displays a statement that no new connections may be established on the SAP.
reference	Identifier for connections associated with this particular transport selector.
peer reference	Identifier for the peer reference associated with this particular transport selector.
peer NSAP address	Identifies the peer NSAP address for each connection (30-40 hexadecimal digits).
peer TSAP selector	Identifies the TSAP selector for each connection (two hexadecimal digits). The identifier may be truncated if the transport is communicating with a foreign OSI transport implementation.

Transport Reference Status Display

The transport reference status display provides the following information about the specified connection.

reference	Identifies the connection to which the display applies.	
protocol stack	Transport class and network service variant.	
connection state	State of the connection, as follows:	
	connection request sent	A connection request has been sent for the identified connection.
	Connection request received	The connection received a connection request. The connection is waiting for acceptance or rejection from the user.
	Confirm acknowledge wait	The user has accepted the connection and has sent a connect confirm TPDU. The user is awaiting acknowledge of the connect confirm.
	Open	The identified connection is open, ready to transmit and receive messages.
	Closing	The identified connection is being disconnected.

`Disconnecting`	The identified connection is being disconnected because the associated SAP is being closed (service shutdown).
`Reference timer wait`	The connection has no more activity. Connection waiting for reference timer to expire.

Displays the following information about the connection:

- Connection table address
- User connection identifier
- Transport selector, which is the identifier of the transport selector associated with the SAP
- Network service priority
- Next assignable transmit sequence number
- Next expected receive sequence number
- Credits granted to the peer
- Credits granted from the peer
- Average round trip (in milliseconds)
- Length data receive queue
- Length normal data queue
- Length of normal ACK queue
- Local acknowledgement timer
- Delayed acknowledgments
- Whether there is expedited data outstanding
- Whether the connection is transport congested
- Whether the connection is user congested
- Whether this is an outgoing or incoming connection
- Whether Transport PDUs are checksummed

DISPLAY_OUTCALL_GATEWAY_STATUS (DISOGS) NETOU Command

Purpose Displays the configuration of the Outcall Gateway. The display contains a header entry that includes the status of the Outcall Gateway itself. Following the header entry are one or more Server entries, each describing information about the servers associated with a particular title.

Format DISPLAY_OUTCALL_GATEWAY_STATUS

Parameters None.

Responses --INFORMATIVE-- Outcall Gateway is not defined.

Examples

```
display_outcall_gateway_status

Outcall Gateway                 : STARTED
Services                        : DEVICE_OUTCALL, PASSTHROUGH
Registration Title              : $OUTCALL
Default Inactivity Timer        : 600
Total Server Connections        : 46
Total Client Connections        : 11

Title                           : ARHBE
Allowed Access                  : PASSTHROUGH
Access Status                   : AVAILABLE
Inactivity Timer                : 300
Control Server Port             : YES
Modem Protocol                  : NONE
Total Server Ports              : 32
Device outcall Servers          :  0/0 (configured/available)
Passthrough Servers             : 32/28 (configured/available)
Idle Message                    : NONE

Title                           : DIALOUT
Allowed Access                  : DEVICE OUTCALL, PASSTHROUGH
Access Status                   : AVAILABLE, AVAILABLE
Inactivity Timer                : 300
Control Server Port             : YES
Modem Protocol                  : HAYES
Total Server Ports              : 2
Device outcall Servers          :  2/1 (configured/available)
Passthrough Servers             :  2/1 (configured/available)
Idle Message                    : NONE

Title                           : VAX
Allowed Access                  : DEVICE OUTCALL, PASSTHROUGH
Access Status                   : AVAILABLE, BUSY
Inactivity Timer                : 600
Control Server Port             : YES
Modem Protocol                  : NONE
Total Server Ports              : 12
Device outcall Servers          : 12/6 (configured/available)
Passthrough Servers             :  4/0 (configured/available)
Idle Message                    : NONE
```

DISPLAY_PASSTHROUGH_STATUS (DISPS) NETOU Command

Purpose Returns the configuration for the Interactive Passthrough Gateway (IPG) and displays the status of all connections the IPG supports.

NOTE

In future releases, this command will no longer be supported. You should start using the DISPLAY_OUTCALL_GATEWAY_STATUS command as soon as possible.

Format **DISPLAY_PASSTHROUGH_STATUS**

Parameters None.

Responses The response provides five lines of header information about the passthrough service, followed by a four-line entry for each unique server connected to the IPG. See example of successful status display.

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>

The following reponses is displayed when the service is not defined.

--ERROR-- Passthrough Service not defined.

Examples

```
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
```

Passthrough Status Display Description

The passthrough status display includes the following information about the passthrough service:

`Default Inactivity Timer`	The value of the default inactivity timer.
`Status`	The operational status of the passthrough gateway.
`Service Title`	The title of the configured passthrough gateway.
`Total Server Connections`	The total number of connections configured to connect to this instance of the gateway.
`Total Client Connections`	The total number of server connections in actual use.

The display provides the following information about each unique server connected to the interactive passthrough gateway:

`Server Title`	The title of the server connected to the IPG, as specified in the DEFPT command.
`Inactivity Timer`	The value of the inactivity timer for this server, as specified in the DEFPT command.
`Server Connections`	The number of server connections associated with this server title.
`Client Connections`	The number of clients currently connected to the server connections of this server title.

DISPLAY_RECORDER_LOG_GROUP (DISRLG) (NOS MDI Only) NETOU Command

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 The following header precedes the display of log groups (see example).

Recorder Log Groups.

If no recorder log groups are defined for the DI, or if log groups are not defined for a specified trunk, the following appropriate 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

```
display_recorder_log_group

Recorder Log Groups for trunk $mci4.  Connection is active.
Log Group                          Priority
CATENET                            1
```

DISPLAY_REMOTE_LOAD_SUPPORT (DISRLS) NETOU Command

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 described in this manual. The returned status information includes the status of the load help provided by the remote systems.

Format **DISPLAY_REMOTE_LOAD_SUPPORT**

Parameters None.

Responses The following header precedes the display (see example).

Remote Load Support Configuration

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.

```
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.

```
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_RIP_SERVICE (DISRS)
NETOU Command

Purpose Displays the current configuration of RIP (Routing Information Protocol) services. This command displays the RIP service information that was configured with ADD_RIP_SERVICE commands. The following information is displayed for each IP network that has the RIP service configured:

- IP Network

  The IP address of the directly connected IP network.

- Cost

  The value of the TRUNK_COST parameter supplied on the ADD_RIP_SERVICE command.

- Trusted neighbors

  The IP address of each *trusted* IP host on this IP network that was configured by the TRUSTED_NEIGHBORS parameter. If no IP addresses were configured as *trusted* IP hosts, ALL is displayed.

- Excluded routes

  The IP address of each network route that was configured by the EXCLUDED_ROUTES parameter. If no routes were configured for exclusion, NONE is displayed.

- Included routes

  The IP address of each network route that was configured by the INCLUDED_ROUTES parameter. If no routes were configured for explicit inclusion, ALL is displayed.

- ARP routing

  Indicates whether ARP proxy routing is enabled or disabled

Format DISPLAY_RIP_SERVICE (DISRS)

Parameters None.

Responses RIP Configuration Display {Display follows, see example}

If no RIP services are configured, the following response replaces the configuration display:

No RIP services are defined for this system.

Examples

```
display_rip_service

                       RIP Configuration Display

  IP          Mode Cost    Trusted       Excluded      Included    ARP
Network                   Neighbors       Routes        Routes    Rtng.

129.179.0.0  BOTH   1     129.179.0.2    129.178.0.0     ALL       OFF
                          129.179.0.5    129.176.0.0
                                         140.150.0.0
129.176.0.0  RECV   1        ALL            NONE         ALL       OFF

129.177.0.0  SEND   2        ALL            NONE       150.1.0.0   OFF
```

DISPLAY_SERVICE_DISPLAY (DISSD) NETOU Command

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 *ALL*
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)

Displays 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 The following header precedes a list of service_name, text, down_text, and temporary_down_text.

Services in the displayable list.
<Service_name>
T: <text>
DT: <down_text>
TDT: <temporary_down_text>

The following header precedes the services you listed on the command, which were not in the displayable services list.

Services not in the displayable list.

<Service_name>

The following header appears if no services were defined.

--ERROR-- No services defined in displayable list.

Remarks Also see the CHANGE_SERVICE_DISPLAY and CHANGE_SERVICE_DISPLAY_TEXT commands in this manual.

Examples

```
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_SLIP_GW (DISSG)
NETOU Command

Purpose Displays the status and entry names for the specified SLIP gateway.

Format

DISPLAY_SLIP_GW
GATEWAY_NAME = *name* or *ALL*

Parameters

GATEWAY_NAME (GW)

Specifies the logical name of the SLIP gateway as previously defined in a DEFINE_SLIP_GW command. If ALL is specified, all SLIP gateways are displayed. The default is ALL.

Responses

```
SLIP gateway <gateway_name>: STATUS = <slip_status>

 Gateway IP Address = <ip_address>      Maximum SLIP Hosts = <1..100>
 Subnet Mask Size = <10..30>            Current Connections = <1..100>
 Inactivity Timeout = <0..65535>        Local Lines Only = <TRUE/FALSE>

 Registered titles
 <title1>
 <title2>
   .

SLIP Host Name = <host_name>
 IP address range = <ip_address>..<ip_address>
 Host name string = <host_name_string>
 Active Connections:
  <ip_address>
```

--ERROR-- SLIP gateway <gateway_name> is not defined.

Examples

```
display_slip_gw gateway_name=slip_gw

SLIP gateway SLIP_GW: STATUS = RUNNING

     Gateway IP Address = 129.179.052.000   Maximum SLIP Hosts = 100
     Subnet Mask Size = 26                  Current Connections = 1
     Inactivity Timeout = INFINITE          Local Lines Only = FALSE

     Registered titles
    SLIP_GW

    SLIP Host Name = SLIPHOST
     IP address range = 129.179.052.062..129.179.052.062
     Host name string = Attaching sneezy (129.179.052.062) to domain CDCNET via
     slip_gw (129.179.052.062)
     Active Connections:
      129.179.052.062

    display_slip_gw gateway_name=slip_ff

    --ERROR-- SLIP gateway slip_ff is not defined.
```

DISPLAY_SNMP_AGENT_OPTIONS (DISSAO) NETOU Command

Purpose Displays information configured using commands DEFINE_SNMP_AGENT, CHANGE_SNMP_AGENT, and ADD_SNMP_COMMUNITY.

Format **DISPLAY_SNMP_AGENT_OPTIONS**

Parameters None.

Responses The SNMP Agent is not defined.

SNMP Agent Options

```
name = <.... name, up to 255 in length ....>
contact = <.... contact, up to 255 in length ....>
location = <.... location, up to 255 in length ....>
max_pdu_size = xxxx

Community                IP Address        Send Traps  Access Mode

'<community name>'       xxx.xxx.xxx.xxx   <boolean>   read-write
'<community name>'       ALL               N/A         read-only

                                           :
```

Examples

```
display_snmp_agent_options

  SNMP Agent Options

  name = ahse.cdc.com
  contact = John Smith, Network Administrator (612) 555-1212
  location = Building 2, 3rd floor
  max_pdu_size = 8192

  Community              IP Address        Send Traps  Access Mode

  'public'               ALL               N/A         read-only
  'manager 1'            128.52.1.12       TRUE        read-write
                         129.25.42.127     FALSE       read-only
  'manager 2'            126.45.78.63      FALSE       read-only
```

DISPLAY_SOFTWARE_LOAD_STATUS (DISSLS) NETOU Command

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

Parameters None.

Responses The following header precedes the load status display (see example).

Software Load Status

Examples

```
display_software_load_status

   Software Load Status
   name                                    retained
   ----------------------------------------------
   diagnostic_me                           no
   dvm_mod_smm                             yes
   get_command_file                        no
   on_line_loader                          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 is 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) NETOU Command

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

Parameters None.

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 replaces the source alarm group display):

No source alarm groups defined.

(If no source alarm messages are defined, the following response replaces the list of alarm message numbers):

No alarm message numbers defined.

Examples

```
display_source_alarms

                    Source alarms
Alarm groups:
    CATENET
Alarm message numbers defined:
    17..82        198        252..280
```

DISPLAY_SOURCE_LOG_GROUP (DISSLG)
NETOU Command

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**

Parameters None.

Responses

Source Log Groups

Log Group <name>
Log message numbers defined:
<list of message numbers>

(Returns a response for each defined log group. If a log group has no enabled messages, the following response replaces the message list for the group):

No log message numbers defined.

(If the DI has no defined source log groups, the following response replaces the log group display):

No source log groups defined.

Examples

```
display_source_log_group

            Source Log Groups
Log Group CATENET
Log message numbers defined:
1..200          300          350
```

DISPLAY_SYSTEM_OPTIONS (DISSO)
NETOU Command

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, see the DEFINE_SYSTEM command description in this manual. Allowed keyword values include the following.

SYSTEM_NAME (SN)
DATA_BUFFER_SIZE (DBS)
BUFFER_PERCENTAGE (BP)
BUFFER_BOUNDARY_PERCENTAGES (BBP)
MEMORY_BOUNDARY_PERCENTAGES (MBP)
MEMORY_MANAGER_PERIOD (MMP)
RESERVED_SYSTEM_SPACE (RSS)
STANDARD_STACK_SIZE (SSS)
DEFAULT_CHANNEL_TRUNK (DCT)
ROUTING_SYSTEM (RS)
CLOCKING_SYSTEM (CS)
MAXIMUM_RECOVERIES (MR)
COMPRESS_DUMP (CD)
ALL (A)

Default is ALL.

Responses The following header precedes the options display (see example).

System options.

Remarks For more information on these parameters, see the DEFINE_SYSTEM command description in this manual.

Examples

```
display_system_options

System options
system_name = engineering_tdi_1
data_buffer_size (before reset) = 144
data_buffer_size (after reset) = 524
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 = C170_trunk
clocking_system = no
maximum_recoveries = 5
compress_dump = yes
```

DISPLAY_TCP_INTERFACE_OPTIONS (DISTIO) NETOU Command

Purpose Displays the TCP parameters defined from the DEFINE_TCP_INTERFACE command (described in this manual).

Format **DISPLAY_TCP_INTERFACE_OPTIONS**

Parameters None.

Responses The following header precedes the options display (see examples).

TCP Interface Options

If the TCP interface has not been defined, the following response is displayed within the options display.

TCP Interface is not defined.

Remarks See the DEFINE_TCP_INTERFACE command for a description of the options to be displayed.

Examples

```
display_tcp_interface_options

TCP Interface Options
ack_percentage = 50
accept_strategy = in_window
max_buffers = 2048
max_connections = 200
max_segment_size = 1510
quiet_time = 20
retransmit_strategy = adaptive
retransmit_time = 3
security_checking = none
time_to_live = 60
syn_ack_time = 60
start = true
```

DISPLAY_TCP_INTERFACE _STATUS (DISTIS) NETOU Command

Purpose Displays the TCP connection status. The display consists of the local and remote IP addresses, the TCP port, the state of the connection and the size in bytes, of the send and receive queue.

Format **DISPLAY_TCP_INTERFACE _STATUS**

Parameters None.

Responses The following header precedes the status display (see examples).

TCP Interface Connection Status

If TCP connection is not defined, the following response replaces the status display.

TCP Interface is not defined.

If no TCP connections exist, the following response is displayed.

TCP Interface Connection Status No Connections.

Examples

```
display_tcp_interface_status

    TCP Interface Connection Status

Local Socket        Remote Socket       State        Send-Q  Rec-Q
192.5.209.1.telnet  192.5.209.53.1025   Established  1000    0
192.5.209.1.ftpc    192.5.209.52.1026   Established  0       0
192.5.209.1.telnet  *.*                 Listen       0       0
192.5.209.1.ftpc    *.*                 Listen       0       0
192.5.209.1.1027    192.5.209.52.ftpc   Established  0       2048
```

TCP Interface Connection Status Display Description

The TCP interface connection status display includes the following information.

Local Socket	The IP address and port number of the Upper Layer Protocol (ULP) that this DI supports.
Remote Socket	The IP address and port number of the peer Upper Layer Protocol (ULP), the other end of the connection.
State	The TCP state of the connection.
Send-Q	The number of bytes currently in TCP send queue for the respective connection.
Rec-Q	The number of bytes currently in TCP receive queue for the respective connection.

DISPLAY_TELNET_SERVER_STATUS (DISTSS) NETOU Command

Purpose Displays the configuration options and status of one or more selected Telnet servers.

Format **DISPLAY_TELNET_SERVER_STATUS**
SERVER_NAME = list of name(s) or *ALL*

Parameters *SERVER_NAME (SN)*

Specifies the name(s) of the servers for which configuration options and status information is displayed. Default is ALL.

Responses The following header precedes the status display (see example).

Telnet Server Status

If the named Telnet server is not defined, the following response replaces the status display.

--WARNING-- TELNET server <server_name> was not found.

If the keyword ALL is specified and no Telnet servers are defined, the following response replaces the status display.

--WARNING-- No TELNET servers were found.

Examples

```
distss server_name=arhnet

Telnet Server Status

Server ARHNET
 IP Address                      : 129.179.52.50
 TCP Port Number                 : 23
 Protocol                        : TELNET
 Terminal Definition Procedure   : NONE
 Terminal User Procedure         : NONE
 Procedure File Option           : NONE
 Connection Connect Timeout      : 120
 Connection Disconnect Timeout   : 120
 User Connection Limit           : 4
 Initial Option Negotiation      : (WILL_SGA, WILL_ECHO)
 Validate Users                  : TRUE
 TCP Timeout                     : 300
 Active Connections              : 12
 Server Status                   : ACTIVE
```

DISPLAY_TELNET_SERVICE_NAME (DISTSN) NETOU Command

Purpose Displays the name of the Telnet service used when accessing Telnet from this DI. This service is accessed when a NOS/VE host application issues a VTP Create CDCNET Connect request to a Telnet service. This is done, for example, when a terminal user executes the CREATE_TELNET_CONNECTION (TELNET) command on a NOS/VE system.

The name must be the title of a User Telnet Gateway configured in the network.

Format **DISPLAY_TELNET_SERVICE_NAME**

Parameters There are no parameters for this command.

Responses The Telnet service name is: <service_name>.

Examples

```
display_telnet_service_name

The Telnet service name: TELNET.
```

DISPLAY_TEST_STATUS (DISTS)
NETOU Command

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 or *keyword value*
STATUS_TYPE= keyword value
DISPLAY_OPTION = keyword value

Parameters *DEVICE_NAME (DN)*

Physical name of a single hardware device or a group keyword value. Specify a hardware device name with a $ character, a board type, and board slot number. Example physical names include $MCI2, $CIM3, $LIM1, $LIM2_PORT1, $ESCI4, $URI3, and NETWORK_1. Specify a group with the appropriate group keyword value. Example group keywords include ALL (A), MAJOR (M), LIMS (L), PORTS (P), or SUBNET (S). Default is ALL.

STATUS_TYPE (ST)

Specifies the type of test status to be returned for the specified devices. The allowable keywords include the following:

Keyword Value	Description
ALL (A)	Displays the status of all tests.
RUNNING (R)	Displays the status of only the running tests.
FAILED (F)	Displays the status of only the failed tests.
EXCEPTION (E)	Displays the status of only the exception tests.

Default is ALL.

DISPLAY_OPTION (DO)

Specifies the format and level of status of the test to be returned in the response.

Keyword Value	Description
SUMMARY (S)	Displays a single line of information per device.
EXPANDED (E)	Displays multiple lines of information per device.

Default is SUMMARY.

Responses The following header precedes the test status information (see examples).

<device> <status> <test_type>test

When errors occur, one of the following responses appears.

Subnet <subnet_name> has not been run. The last reset was at <date> <time>.

--ERROR-- Device <device_name> not installed in system.

--ERROR-- No RUNNING devices found.

--ERROR-- No FAILED devices found.

--ERROR-- No EXCEPTION devices found.

--ERROR-- Subnet <subnet_name> is unknown.

Examples The following example shows a status display for an online test that is currently running.

```
display_test_status device_name=$lim1_port2 do=expanded

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.

```
display_test_status device_name=$cim5

                $CIM5       FAILED  on-line  test.

                $CIM5       FAILED  on-board test.

display_test_status device_name=$cim5 do=expanded

                CIM test status
                CIM slot number= 5
                FAILED  on-line version 10H1 01/24/85 14.43.31
                Testing CIM/SMM interface
                pass count= 5  total errors= 1

                CIM test status
                CIM slot number= 1
                FAILED  on-board version 09A1 01/24/85 14.43.31
```

Displays for device-passing tests:

```
display_test_status device_name=$mci1

MCI test status
MCI slot number = 1
PASSED on-board version 08H1 01/16/86 14.34.21

display_test_status device_name=$esci3 do=summary

$ESCI3     PASSED  on_line test.

or

$ESCI3     PASSED  on_board test.

display_test_status device_name=$esci3 do=expanded


ESCI test status
ESCI slot number= 3
PASSED  on-line version 2301 01/16/87 14.34.21
pass count= 10

or

ESCI test status
ESCI slot number= 3
PASSED  on-board version 0806 01/16/87 14.34.21
```

In the following two examples, multiple devices were tested with online and inline diagnostics.

```
display_test_status status_type=exception do=summary

$LIM7_PORT2  RUNNING on-line  test
$ESCI4       RUNNING on-line  test
$LIM7_PORT5  FAILED  on-line  test
$MCI7        FAILED  in-line  test
$NET_1 RUNNING subnet connect test

display_test_status dn=major st=exception do=summary

$ESCI4       RUNNING on-line  test.
$MCI7        FAILED  in-line  test
```

In this example, a CIM online test passed on 01/16/85, a LIM online test failed on 03/18/85, a PORT online test failed 04/28/85, and a URI online test failed on 04/29/85. 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.

```
display_test_status device_name=$cim1 do=summary

$CIM1      PASSED  on-line  test.  Lim/Port errors detected.

display_test_status device_name=$cim1 do=expanded

CIM test status
CIM slot number = 1
PASSED on-line version 10H1 01/16/85 14.34.21
pass count = 10
  LIM/PORT failure summary:
    FAILED lim 4        on-line  version 10H1 03/18/85 04.18.01
    FAILED lim 5 port 1 on-line  version 10H1 04/28/85 10.21.12
    FAILED uri 6        on-line  version 2301 04/29/85 07.55.20
```

In this example, a LIM onboard test passed on 01/16/85, and a PORT online test failed on 01/24/85 and 01/25/85. 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.

```
display_test_status device_name=$lim3 do=summary

$LIM1      PASSED  on-board test.  Port errors detected.

display_test_status device_name=$lim3 do=expanded

LIM test status
CIM slot number = 6
LIM slot number = 3
PASSED on-board version 10H1 01/16/85 14.34.21
  PORT failure summary:
    FAILED port 1 on-line  version 10H1 01/24/85 14.34.21
    FAILED port 3 on-line  version 10H1 01/25/85 10.21.12
```

In this example, an onboard diagnostic issued a PASSED response following a nonfatal error. The ESCI transceiver has failed during onboard execution. This is a nonfatal error.

```
display_test_status device_name=$esci3 do=summary

$ESCI3     PASSED  on_board test.  Transceiver errors detected.

display_test_status device_name=$esci3 do=expanded

ESCI test status
ESCI slot number= 3
PASSED  on-board version 0806 01/16/87 14.34.21
Transceiver errors detected.
```

In this example, the last inline diagnostic on this device PASSED.

```
display_test_status device_name=$mci1 do=expanded

MCI test status
MCI slot number= 1
PASSED  in-line version 2601 04/16/86 14.34.21
```

In this example, the last inline diagnostic on this device FAILED.

```
display_test_status device_name=$mci3 do=summary

$MCI3       FAILED  in_line  test.

display_test_status device_name=$mci3 do=expanded

MCI test status
MCI slot number= 3
FAILED  in-line version 2601 04/16/85 14.53.31
pass count= 1  total errors= 8
```

In the following two examples, the inline subnet test PASSED.

```
display_test_status device_name=$net_1 do=summary

$NET_1 PASSED subnet connect test

display_test_status device_name=$net_1 do=expanded

$NET_1 PASSED subnet connect test
Network type = Ethernet
Network device = $ESCI5
Test Date/Time = 03/01/89 15:22:17
Destination = TDI_C2
Message Length = MIXED
Message Count = 20
Message Interval (msec.)= 200
repeat Pass = 2
Total Messages sent = 40
Total Bytes sent = 18921
Total Messages Rcvd = 40
Total Bytes Rcvd = 18921
Pass count = 2
```

In the following two examples, the command entry and subnet test FAILED.

```
display_test_status device_name=$net_2 do=summary

$NET_1 FAILED subnet connect test

display_test_status device_name=$net_2 do=expanded

$NET_1 FAILED subnet connect test
Network type = Ethernet
Network device = $ESCI5
Test Date/Time = 02/22/89 09:12:37
Destination = TDI_C2
Message Length = MIXED
Message Count = 20
Message Interval (msec.)= 0
repeat Pass = 0
Total Messages sent = 20
Total Bytes sent = 9342
Total Messages Rcvd = 12
Total Bytes Rcvd = 4921
Pass count = 1
Total Errors = 1
Buffer Length = 217
Data checksum error in a returned message
```

Diagnostic Test Status Display Description

The diagnostic test status display provides either a summary or expanded level of information about the status of device diagnostics. The summary level provides a single display line reporting the status of the diagnostics on each device or subnet. The expanded level provides multiple display lines reporting the status of the diagnostics on each device or subnet, as well as other information.

The summary test status display includes the following information (as appropriate for the device being tested).

Device name

Physical name of the tested device

Status of last test on device, including the following:

Test status (RUNNING, PASSED, FAILED or blank). The field is blank no test has run on the device since the DI was last reset.

Test type (on-board, online, or inline).

- An indication that a subassembly has errors or that nonfatal errors exist.

The summary display for a subnet includes the following information:

Physical name of the subnet (Subnet_name)

Test status (RUNNING, PASSED, or FAILED)

Date/time stamp of the last test activity. If a test has not been run on this subnet, the summary line shows the date and time of the last reset.

The expanded 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 online 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.

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.

The expanded display for a subnet includes the summary line containing the subnet name, status and date/time stamp, as well as the following information:

Network type and the physical network device name

Parameters from the command input:

Destination system.

Message length.

Message count.

Message interval.

Repeat pass count.

Total number of actual messages sent/received.

Total number of actual bytes sent/received.

Actual number of passes completed.

For a test that has FAILED, the following additional information is displayed:

Total error count.

Buffer length of the failed message.

An additional error message indicating the cause of the failure.

DISPLAY_TRACE_PROCESS (DISTP) NETOU Command

Purpose Displays trace information collected for the given process. Prior to execution of this command, the trace must be started for the specified process. For more information regarding the initiation of a trace process, see the START_TRACE_PROCESS command.

Format

DISPLAY_TRACE_PROCESS
PROCESS = name
DISPLAY_SIZE = integer 1 .. 65535
FIXED_DISPLAY_OPTION = keyword value
VARIABLE_DISPLAY_OPTION = keyword value

Parameters

PROCESS (P)

Specifies the name of the process to be displayed. For names of processes that can be currently traced, see the START_TRACE_PROCESS command.

DISPLAY_SIZE (DS)

Specifies the number of trace entries to be displayed. See examples for further detail. The default is 5.

FIXED_DISPLAY_OPTION (FDO)

Specifies how the fixed trace information is to be displayed. The default is PROCESS_SPECIFIC. The allowed keyword values are:

HEXDATA (H)

Displays the fixed display information as a contiguous line of hexadecimal numbers.

DISPLAY_MEMORY (DM)

Displays the fixed display information as hexadecimal numbers with 16 bytes per line, a space separating every 2 bytes, and the character representation displayed at the right.

PROCESS_SPECIFIC (PS)

Displays the fixed display information with the received data being interpreted.????

VARIABLE_DISPLAY_OPTION (VDO)

Specifies how the variable trace information is to be displayed. The default is PROCESS_SPECIFIC. The allowed keyword values are:

HEXDATA (H)

Displays the variable display information as a contiguous line of hexadecimal numbers.

DISPLAY_MEMORY (DM)

Displays the variable display information as hexadecimal numbers with 16 bytes per line, a space separating 2 bytes, and the character representation displayed at the right.

PROCESS_SPECIFIC (PS)

Displays the variable display information with the received data being interpreted.

Responses

Specific displays vary according to the process specified (see the examples).

--ERROR-- TRACE entry point not found for process <process>.

The event codes in the displays have the following meanings.

All Event Processes:

Code	Meaning
0	Unused/uninitialized
1	Protocol Data Unit (PDU) transmitted to lower layer
2	PDU received from lower layer
3	Timer expired
4	Workcode received
5	Link up status received from 3A
6	Link down status received from 3A
7	Buffer state congested
8	Buffer state uncongested

DFA Processes:

Code	Meaning
20	Bad file control block received (for example, could occur if a timeout occurs before Transport Indication)

Directory Processes:

Code	Meaning
20	Data store entry added to hash bucket
21	Data store entry added to timer chain
22	Data store entry deleted from hash bucket
23	Data store entry deleted from timer chain
24	Broadcast version 3 PDU transmitted

ES-IS Routing Processes:

Code	Meaning
20	Extension PDU received
21	Extension PDU transmitted
22	Query request sent
23	Query response transmitted
24	ES-only notification
25	Initialization of IS-IS complete

HDLC Processes:

Code	Meaning
20	PDU in extended mode received from lower layer
21	PDU to be discarded
22	PDU released due to not sequenced correctly
23	PDU released due to bad next sequence number
24	PDU released due to action=discard input
25	PDU released due to either (1) frame aborted, (2) cre error, (3) buffer underrun
26	PDU released due to header length too small
27	PDU released due to extended header length too small
28	Mode change: was normal mode, now in extended mode

RIP Processes:

Code	Meaning
20	Update an existing IPSR network
21	Add a new network entry to the IPSR routing tables
22	Delete an existing IPSR network entry
23	Delete an ARP-owned entry and create a RIP-owned entry
24	RIP entry added due to ADD_RIP_SERVICE command processor
25	RIP entry deleted due to one of the following causes: the DELETE_RIP_SERVICE command processor, UDP closes its RIP SAP, or the IP net for which RIP service is configured is cancelled.

Transport Processes:

Code	Meaning
20	Transmit a CLNS NSDU (Network Service Data Unit)
21	Receive a CLNS NSDU
22	Transmit a CONS NSDU
23	Receive a CONS NSDU

Examples

```
display_trace_process system

TRACE display at 07:44:12.938 for process SYSTEM, 45 events traced.

Entry 41, event code 0, occurred 13200 milliseconds ago
Fixed info =
TCB Address = 106804   SYSTEM_AUDITS
    Status Register=0011    Time Slices=0       ITMs Queued=0
Variable info =
    TRACEBACK FROM          105785E   IL_CHECKSUM_PMM+ 16
       CALLED FROM          100F0B4   OLL_PROGRAM_INTERFACE_PROCS+ 1A6
       CALLED FROM          1015C0E   SYSTEM_AUDITS+ 85C
       CALLED FROM          1015780   SYSTEM_AUDITS+ 3CE
       CALLED FROM          10106D6   OLL_PROGRAM_INTERFACE_PROCS+ 17C8

Entry 42, event code 0, occurred 13100 milliseconds ago
Fixed info =
TCB Address = 106804   SYSTEM_AUDITS
    Status Register=0008    Time Slices=0       ITMs Queued=0
Variable info =
    TRACEBACK FROM          1057860   IL_CHECKSUM_PMM+ 18
       CALLED FROM          100F0B4   OLL_PROGRAM_INTERFACE_PROCS+ 1A6
       CALLED FROM          1015C0E   SYSTEM_AUDITS+ 85C
       CALLED FROM          1015780   SYSTEM_AUDITS+ 3CE
       CALLED FROM          10106D6   OLL_PROGRAM_INTERFACE_PROCS+ 17C8

Entry 43, event code 0, occurred 13000 milliseconds ago
Fixed info =
```

```
TCB Address = 106804   SYSTEM_AUDITS
   Status Register=2004    Time Slices=0       ITMs Queued=0
Variable info =
   TRACEBACK FROM          100816A   EXEC_PMM+ 164
      CALLED FROM          1015CB8   SYSTEM_AUDITS+ 906
      CALLED FROM          1015780   SYSTEM_AUDITS+ 3CE
      CALLED FROM          10106D6   OLL_PROGRAM_INTERFACE_PROCS+ 17C8

Entry 44, event code 0, occurred 12900 milliseconds ago
Fixed info =
TCB Address = 106804   SYSTEM_AUDITS
   Status Register=0000    Time Slices=0       ITMs Queued=0
Variable info =
   TRACEBACK FROM          1057860   IL_CHECKSUM_PMM+ 18
      CALLED FROM          100F0B4   OLL_PROGRAM_INTERFACE_PROCS+ 1A6
      CALLED FROM          1015C0E   SYSTEM_AUDITS+ 85C
      CALLED FROM          1015780   SYSTEM_AUDITS+ 3CE
      CALLED FROM          10106D6   OLL_PROGRAM_INTERFACE_PROCS+ 17C8

Entry 45, event code 0, occurred  5200 milliseconds ago
Fixed info =
TCB Address = 1C4868   LOCAL_CONSOLE_FORMATTER
   Status Register=0009    Time Slices=0       ITMs Queued=0
Variable info =
   TRACEBACK FROM          100A310   CYBIL_RUN_TIME_ROUTINES+PMM+ 76E
      CALLED FROM           1B1266   CDCNET_DI_MESSAGE_FORMATTER+ 1FB2
      CALLED FROM           1B08A4   CDCNET_DI_MESSAGE_FORMATTER+ 15F0
      CALLED FROM           1AE812   LOCAL_CONSOLE+_FORMATTER+ 118
      CALLED FROM          10106D6   OLL_PROGRAM_INTERFACE_PROCS+ 17C8


display_trace_process process=dod_internet ..
  display_size=3 fixed_data_option=hexdata ..
  variable_display_option=process_specific

TRACE display at 15:07:14.226 for process DOD_INTERNET, 95 events traced.

Entry 93, event code 2, occurred 19500 milliseconds ago
Fixed info = 0800253006B9080025300688080014140300080005010001001CC3C0001DBC3C
             Variable info =
                 IP header = 450000290031000039060FEBF81B33FCF81B33FA9
                 Source = 129.179.63.207, Destination = 129.179.63.169
                 ID = 0031(16), Time-to-live = 57, Protocol = 6, PDU length = 41
                 IP User Data =
                       0  0472 0017 0271 0A32 0196 28F9 5018 1000      r   q 2  ( P
                      10  C0E2 004F 2036 3630 0025                       0 660 %

             Entry 94, event code 1, occurred 19500 milliseconds ago
             Fixed info = 001A44220191080025300688382481B33FCF6B7E0293000161636B7E00271EB2
             Variable info =
                 IP header = 450000290028000038060FCC581B33FA981B33FCF
                 Source = 129.179.63.169, Destination = 129.179.63.207
                 ID = 002B(16), Time-to-live = 59, Protocol = 6, PDU length = 41
                 IP User Data =
                       0  0017 0472 0196 28F9 0271 0A33 5018 1000      r    (  Q 3PP
                      10  C104 002C 20                                  ,

             Entry 95, event code 2, occurred 19400 milliseconds ago
             Fixed info = 0800253006B9080025300688080FEFE0300080005010001001CC3C000143FCC
             Variable info =
                 IP header = 450000280032000039060FEBF81B33FCF81B33FA9
                 Source = 129.179.63.207, Destination = 129.179.63.169
                 ID = 0032(16), Time-to-live = 57, Protocol = 6, PDU length = 40
                 IP User Data =
                       0  0472 0017 0271 0A33 0196 28FA 5010 1000      r   q 3  ( P
                      10  0304 DE34 3630 3630 0025                       46060 %


display_trace_process process=X25_PL ..
  display_size=3 fixed_data_option=display_memory ..
  variable_display_option=display_memory

TRACE display at 14:46:03.190 for process X25_PL , 24013 events traced.

Entry 9 , event code 1, occurred 24900 milliseconds ago
Fixed info =
       0  0022 B516 0021 DD1C 3120 2020 2020 2020    "   !  1
      10  2020 2020 2020 2012 0016 0016 0004 0000
      20  0000 0000 0000 0000 0000 0000 0005 0001
      30  0000 0002 0001 0002
Variable info =
       0  1016 2814 1400 0304 0000 0AA1 0000 0104     (
      10  0800 2510 0072 0000 0091 0900 25FF FFFF     %  r      %
      20  0000 1056 0800 2510 0072 0000 0091 0900      V  %  r
```

```
      30  25FF FFFF 0000 2749 0800 2510 0072 06FA   %     'I  %  r
      40  0011 0900 25FF FFFF 0000 3191 0800 2510       %     1   %
      50  0072 000A 0011 0900 25FF FFFF              r      %

Entry 10 , event code 1, occurred 23800 milliseconds ago
Fixed info =
       0  0022 B516 0021 DD1C 3120 2020 2020 2020    "   !  1
      10  2020 2020 2020 203E 0016 0016 0004 0000          >
      20  0000 0000 0000 0000 0000 0000 0006 0001
      30  0000 0002 0001 0002
Variable info =
       0  1016 2A14 1400 0304 0000 7B86 0000 0104     *       {
      10  0800 2530 00C1 0000 0091 0900 25FF FFFF     %0        %
      20  0000 1056 0800 2530 00C1 0000 0091 0900      V  %0
      30  25FF FFFF 0000 2749 0800 2530 00C1 06FA   %     'I  %0
      40  0011 0900 25FF FFFF 0000 3191 0800 2530       %     1   %0
      50  00C1 000A 0011 0900 25FF FFFF                    %
```

```
Entry 11 , event code 2, occurred 23800 milliseconds ago
Fixed info =
       0  0022 B516 0021 DD1C 3120 2020 2020 2020    "   !  1
      10  2020 2020 2020 2046 0016 0016 0004 0000          F
      20  0000 0000 0000 0000 0000 0000 0006 0001
      30  0000 0002 0001 0002
Variable info =
       0  1016 A214 1400 0304 0000 7B86 0000 0104             {
      10  0800 2530 00C1 0000 0091 0900 25FF FFFF     %0        %
      20  0000 1056 0800 2530 00C1 0000 0091 0900      V  %0
      30  25FF FFFF 0000 2749 0800 2530 00C1 06FA   %     'I  %0
      40  0011 0900 25FF FFFF 0000 3191 0800 2530       %     1   %0
      50  00C1 000A 0011 0900 25FF FFFF                    %

Entry 12 , event code 1, occurred 600 milliseconds ago
Fixed info =
       0  0022 B516 0021 DD1C 3120 2020 2020 2020    "   !  1
      10  2020 2020 2020 2082 0016 0016 0004 0000
      20  0000 0000 0000 0000 0000 0000 0007 0002
      30  0001 0005 0002 0005
Variable info =
       0  1016 4CFE FE00 821E 0100 0400 7D00 0014     L          }
      10  3984 0F01 BB7B 0000 0000 0031 9108 0025   9    {     1   %
      20  1000 7200                                     r

Entry 13 , event code 1, occurred 600 milliseconds ago
Fixed info =
       0  0022 B516 0021 DD1C 3120 2020 2020 2020    "   !  1
      10  2020 2020 2020 2096 0016 0016 0004 0000
      20  0000 0000 0000 0000 0000 0000 0000 0002
      30  0001 0005 0002 0005
Variable info =
       0  1016 4E18 1800 0000 2401 8000 0000 007D     N     $       }
      10  0000 0559 0000 0014 0004 2749 06FA 3191      Y      'I  1
      20  000A 0104 0000 1056 0000                         V
```

The following are outputs shown in the three different formats for FIXED_DISPLAY_OPTION:

```
display_trace_process process=channel_connection ..
  display_size=1 fixed_display_option=process_specific

TRACE display at 13:43:22.626 for process CHANNEL_CONNECTION, 73 events traced.

Entry 73 , event code 2 , occurred 8766900 milliseconds ago
Fixed info =
 Channel connection cepid = 001CEEF2
 Actual user data size = 0
 Channel Connection PDU header = 0500000000000010000106412E485001
 LIB = 00107E22
Variable info = NONE


display_trace_process process=channel_connection ..
  display_size=1 fixed_display_option=display_memory

TRACE display at 13:44:03.641 for process CHANNEL_CONNECTION , 73 events traced.

Entry 73 , event code 2 , occurred 8807900 milliseconds ago
Fixed info =
       0  001C EEF2 0000 0500 0000 0000 0010 0001
      10  0641 2E48 5001 0010 7E22                  A.HP   ~"
Variable info = NONE


display_trace_process process=channel_connection ..
  display_size=1 fixed_display_option=hexdata

TRACE display at 13:43:52.306 for process CHANNEL_CONNECTION , 73 events traced.

Entry 73 , event code 2 , occurred 8796600 milliseconds ago
Fixed info = 001CEEF2000005000000000000100001064 12E48500100107E22
Variable info = NONE
```

DISPLAY_TRUNK_METRICS (DISTM) NETOU Command

Purpose Displays the statistics collected for one or more communications trunks. The Stream Service Routines (SSRs) supporting the trunks collect the statistics.

Statistics collection for the specified trunks must have been started before executing this command. Use the START_TRUNK_METRICS command, described in this manual, to begin statistics collection by the supporting SSR.

The collected statistics information clears each time trunk metrics for the specified trunks are reported on the network logs. Thus, the displayed statistics are the accumulation of statistics information since the last reporting of trunk metrics. Specify the reporting interval with a parameter on the START_TRUNK_METRICS command. See the START_TRUNK_METRICS command for more information regarding the reporting of trunk metrics.

Format **DISPLAY_TRUNK_METRICS**
TRUNK_NAME = list 1..15 of name or *ALL*

Parameters *TRUNK_NAME (TN)*

Specifies the trunk or trunks for which the statistics are to be displayed. Default is ALL.

Responses Displays vary according to the type of trunks specified (see examples).

One of the following responses displays if the specified metrics are not reported.

Trunk <name> is not defined, or is defined but not started.

SUMMARY metrics have not been started for TRUNK <name>.

Examples This example shows the display for a CHANNEL trunk.

```
display_trunk_metrics mci_trunk

Summary Trunk Metrics

CHANNEL Trunk = MCI_TRUNK

Start of collection     = 14:44:13    Elapsed time            = 00:07:08

Messages transmitted    = 151         Messages received       = 236
Characters transmitted  = 13483       Characters received     = 16474
Transmit rate (bps)     = 248         Receive rate (bps)      = 304
```

```
Message Size Distribution
 size    <256    <512    <768    <1024   <1260   <1536   <1792   1792+
 xmt     145     2       4       0       0       0       0       0
 rcv     236     0       0       0       0       0       0       0

Messages retransmitted = 0          Messages discarded     = 0
Length errors          = 0          Bus errors             = 0
```

This example shows a display for an ETHER trunk.

```
display_trunk_metrics $esci5

Summary Trunk Metrics

ETHER Trunk = $ESCI5

Start of collection    = 14:45:58    Elapsed time           = 00:05:23

Messages transmitted   = 157         Messages received      = 136
Characters transmitted = 10941       Characters received    = 11616
Transmit rate (bps)    = 264         Receive rate (bps)     = 280

Message Size Distribution
 size    <256    <512    <768    <1024   <1280   <1536
 xmt     157     0       0       0       0       0
 rcv     133     1       2       0       0       0

Message Transmission Errors          Message Reception Errors
 too many collisions   = 0           CRC errors             = 0
 lost carrier sense    = 0           alignment errors       = 0
 lost CTS signal       = 0           no resources           = 0
 memory underrun       = 0           memory overrun         = 0
 rejected by DVM       = 0           minimum size error     = 0
 pack message failed   = 0           exceed max length      = 0
 total collisions      = 0           incorrect status       = 0
                                     incorrect length       = 0
```

This shows the display for all the trunks on a system. Note that metrics had not been previously started for the ETHER trunk, but had been started for the HDLC and X.25 trunks.

```
display_trunk_metrics

Summary Trunk Metrics

SUMMARY metrics have not been started for TRUNK $ESCI5.

HDLC Trunk = HDLC_TRUNK

Start of collection    = 14:48:34    Elapsed time           = 00:03:06

I frames transmitted   = 60          I frames received      = 22
S frames transmitted   = 22          S frames received      = 57
U frames transmitted   = 0           U frames received      = 0
Characters transmitted = 3234        Characters received    = 1244
Transmit rate (bps)    = 136         Receive rate (bps)     = 48

Message Size Distribution
 size    <256    <512    <768    <1024   <1280   <1536
 xmt     60      0       0       0       0       0
 rcv     22      0       0       0       0       0

Bad frames recieved    = 0           Frames out of sequence = 0
Frames retransmitted   = 0           Frames discarded       = 0
CRC Errors             = 0

X.25 Trunk = REMOTE_TRUNK

Start of collection    = 14:48:45    Elapsed time           = 00:02:55

I frames transmitted   = 53          I frames received      = 49
S frames transmitted   = 49          S frames received      = 40
U frames transmitted   = 0           U frames received      = 0
Characters transmitted = 3070        Characters received    = 2151
Transmit rate (bps)    = 136         Receive rate (bps)     = 96

Message Size Distribution
 size    <256    <512    <768    <1024   <1280   <1536
 xmt     53      0       0       0       0       0
 rcv     49      0       0       0       0       0

Bad frames recieved    = 0           Frames out of sequence = 0
Frames retransmitted   = 0           Frames discarded       = 0
CRC Errors             = 0
```

DISPLAY_USER_VALIDATION (DISUV)
NETOU Command

Purpose Displays if validation is defined, the default validation domain, and the number of unsuccessful login attempts allowed.

Format DISPLAY_USER_VALIDATION

Parameters None.

Responses User validation is not defined.

User validation is defined.
default_validation_domain = <validation domain>
maximum_login_attempts = <attempts>

Examples

```
display_user_validation

User validation is defined.
default_validation_domain = ARH
maximum_login_attempts = 3
```

DISPLAY_X25_ASYNCTIP_OPTIONS (DISXAO) NETOU Command

Purpose Displays the configuration of an X.25 asynchronous TIP for each X.25 trunk supported by the TIP. The display returns the set of default TIP parameters and terminal usage parameters that apply to each X.25 trunk.

Format

DISPLAY_X25_ASYNCTIP_OPTIONS
TRUNK_NAME = list 1..32 of name
DISPLAY_OPTION = list of keyword values

Parameters

TRUNK_NAME (TN)

Specifies the name(s) of the one or more X.25 trunks for which X.25 asynchronous TIP support is defined. If TRUNK_NAME is not specified, the display presents the X.25 asynchronous TIP definition for each X.25 trunk supported by the TIP.

DISPLAY_OPTION (DO)

Specifies the name(s) of one or more X.25 asynchronous TIP attributes for display. These attributes are defined by parameters on the DEFXA command that configured the line. For more information on these parameters, see the DEFXA command description in this manual. The allowed keyword values are:

TRUNK_NAME (TN)
TERMINAL_DEFINITION_PROCEDURE (TDP)
TERMINAL_USER_PROCEDURE (TUP)
PROCEDURE_FILE_OPTION (PFO)
LOGICAL_CHANNEL_CONCATENATION (LCC)
CALLED_DTE_CONCATENATION (CDC)
CALL_DATA_PROCEDURE (CDP)
CALLED_DTE_ADDRESS_RANGE (CDAR)
CONNECTION_CONNECT_TIMEOUT (CCT)
CONNECTION_DISCONNECT_TIMEOUT (CDT)
USER_CONNECTION_LIMIT (UCL)
ACCEPT_REVERSE_CHARGES (ARC)
PERMANENT_VIRTUAL_CIRCUITS (PVC)
PVC_CHANNEL_CONCATENATION (PCC)
PVC_PACKET_SIZE (PPS)
PVC_WINDOW_SIZE (PWS)
VALIDATE_USERS (VU)
ALL (A)

Default is ALL.

Responses The following header precedes the options display (see examples).

X.25 Asynctip options

Within the options display, the following response is inserted if

support is not defined for a TRUNK_NAME.

X.25 AsyncTip support not defined for trunk <name>.

Within the options display, the following response is inserted if no X.25 asynchronous TIP support is defined in the system.

No X.25 AsyncTip support is defined for this system.

Examples display_x25_asynctip_options

```
X.25 Asynctip options
trunk_name = TELENET_1
terminal_definition_procedure = none
terminal_user_procedure = TELENET_721
procedure_file_option = LOGICAL_CHANNEL_CONCATENATION
called_dte_address_range = 1..15
connection_connect_timeout = 120
connection_disconnect_timeout = 120
user_connection_limit = 4
accept_reverse_charges = no
permanent_virtual_circuits = 1..15
pvc_channel_concatenation = no
pvc_packet_size = 256
pvc_window_size = 3

trunk_name = TELENET_2
terminal_definition_procedure = none
terminal_user_procedure = TELENET_VT100
procedure_file_option = none
called_dte_address_range = 1..15
connection_connect_timeout = 50
connection_disconnect_timeout = 20
user_connection_limit = 4
accept_reverse_charges = yes
permanent_virtual_circuits = 0
pvc_channel_concatenation = no
pvc_packet_size = 0
pvc_window_size = 0
validate_users = yes
```

DISPLAY_X25_GW_INCALL_OPTIONS (DISXGIO) NETOU Command

Purpose Displays the incall validations to be performed by an X.25 application-to-application (A-to-A) gateway. The display includes the DTE addresses, service titles, and reverse charging option accepted by the gateway.

Format

DISPLAY_X25_GW_INCALL_OPTIONS
GATEWAY_NAME=list 1..15 of name or *ALL*

Parameters *GATEWAY_NAME (GN)*

Specifies the names of the X.25 gateways which receive incoming X.25 A-to-A calls. Default is ALL.

Responses --ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <name> is not defined.

Examples

```
display_x25_gw_incall_options gateway_name = X25GW

X.25 Gateway incall configuration
gateway_name: X25GW

remote dte      allowed_titles:          accept
addresses:                               reverse
                                         charging:
408123          PTFS$*                      YES
                QTFS$*                      YES
408456          PTFS$*                      YES
                QTFS$*                      YES
612789          PTFS$*                      YES
                QTFS$*                      YES
OTHER           ALL                         NO
```

DISPLAY_X25_GW_OPTIONS (DISXGO) NETOU Command

Purpose Displays the configuration of an X.25 transparent gateway. The display returns CDCNET title(s) by which the gateway is known in the CDCNET network and the protocol identifiers (prid) used by the gateway to distinguish incoming connections to NOS or NOS/VE hosts.

Format

DISPLAY_X25_GW_OPTIONS
GATEWAY_NAME=list 1..15 of name or *ALL*
DISPLAY_OPTION = list of keyword values

Parameters

GATEWAY_NAME (GN)

Specifies the name(s) of the X.25 gateways which receive incoming X.25 application-to-application calls. Default is ALL.

DISPLAY_OPTION (DO)

Specifies one or more of the gateway attributes for display. The keyword values include the following items.

GATEWAY_NAME (GN)
TRUNK_NAME (TN)
NOS_PROTOCOL_ID (NPI)
CDCNET_PROTOCOL_ID (CPI) or VE_PROTOCOL_ID (VPI)
NOS_PORT_NUMBER (NPN)
VE_OUTCALL_TITLE (VOT)
DTE_ADDRESSING_PROTOCOL (DAP)
ALL (A)

Default is ALL.

Responses

--ERROR-- An X.25 gateway is not defined.

--ERROR-- X.25 gateway <name> is not defined.

Examples

```
display_x25_gw_options

X.25 Gateway options
gateway_name = TELENET_GW
trunk_name = (TELENET_1,TELENET_2)
nos_protocol_id = (0C0(16),0C1(16))
cdcnet_protocol_id = 0C2(16)
dte_addressing_protocol_id = 002(16)
nos_port_number = (023(16),024(16))
ve_outcall_title = TELENET_GW
```

DISPLAY_X25_GW_OUTCALL_OPTIONS (DISXGOO) NETOU Command

Purpose Displays the outcall titles of an X.25 transparent gateway.

Format **DISPLAY_X25_GW_OUTCALL_OPTIONS**
GATEWAY_NAME = list 1..15 of name or *ALL*

Parameters *GATEWAY_NAME (GN)*

Specifies the names of the X.25 gateways to display that provide access to remote applications. Default is ALL.

Responses The following header precedes the X.25 gateway_name display (see examples).

X.25 Gateway outcall configuration.

One of the following responses appears when errors occur.

--ERROR-- X.25 gateway <name> is not defined.

--ERROR-- An X.25 gateway is not defined.

Examples

```
display_x25_gw_outcall_options

gateway_name = Telenet
title = PTFS$FOREIGN
remote_dte_address = 3401
facilities = NONE
protocol_id = c2
user_data = NONE
local_dte_address = NONE
```

DISPLAY_X25_GW_PVC_OPTIONS (DISXGPO) NETOU Command

Purpose Displays the PVC and other pertinent information associated with the given X.25 gateway.

Format **DISPLAY_X25_GW_PVC_OPTIONS**
GATEWAY_NAME = list 1..15 of name or *ALL*

Parameters *GATEWAY_NAME (GN)*

Specifies the names of the X.25 gateways for which PVC information is being requested. Default is ALL.

Responses The following header precedes the X.25 gateway_name display (see examples).

X.25 Gateway PVC configuration

One of the following responses appears when errors occur.

--ERROR-- X.25 gateway <name> is not defined.

--ERROR-- An X.25 gateway is not defined.

--INFORMATIONAL-- No PVCs exist for X.25 gateway <name>.

Examples

```
display_x25_gw_pvc_options

            X.25 Gateway PVC configuration

            gateway_name= Telenet
            title= PVC$FRN
            PVC =100
            pvc_packet_size = 128
            pvc_window_size = 2

            gateway_name= Telenet
            title= PVC$GER
            PVC =101
            pvc_packet_size = 128
            pvc_window_size = 2
```

DISPLAY_X25_INCALL (DISXI) NETOU Command]

Purpose Displays configured X.25 incall definition entries added by using the ADD_X25_INCALL command, for one or more trunk names and one or more remote DTE addresses.

Format

DISPLAY_X25_INCALL
TRUNK_NAME = list 1..32 of name
REMOTE_DTE_ADDRESSES = list 1..31 of string 1..31

Parameters *TRUNK_NAME (TN)*

Specifies the set of trunks associated with the X.25 incall definition entries to be displayed. The value ALL is used to display the specified entries associated with the remote DTE addresses on all trunks previously configured with X.25 incall definition entries. The default is ALL.

REMOTE_DTE_ADDRESSES (RDA)

Specifies the set of remote DTE addresses associated with the X.25 incall definition entries to be displayed. Wild card attributes (as specified by command ADD_X25_INCALL) must exactly match an entry as specified by ADD_X25_INCALL. For example, a value of '123*' will match only on entry '123*'. It will not match all entries with '123' as the first three digits. If a REMOTE_DTE_ADDRESSES value has multiple entries associated with it (due to different values of DTE_ADDRESS_PROTOCOL_ID), all of those tries will be displayed. The value OTHER corresponds to any entries def d with the value OTHER. The value ALL is used to display all X.25 incall definition entries for the trunks specified. The default is ALL.

Responses The configuration for each trunk is displayed separately. For each remote DTE address for which an X.25 incall definition entry is configured, the configurable parameters are displayed. All entries with the same parameter settings are grouped together. Please see the example for the format of this display.

The following response is generated if no X.25 incall definition entry is configured, regardless of input errors:

No X.25 incall definition entry for any trunk is defined.

The following response is generated if X.25 incall definition entries are configured, but none for a trunk explicitly specified:

No X.25 incall definition entry for trunk <trunk_name> is defined.

Error responses:

--ERROR-- An X.25 incall definition entry for the remote DTE address <remote_dte_address> is not defined for trunk <trunk_name>.

--ERROR-- Trunk name <trunk_name> is not defined.

--ERROR-- Trunk name <trunk_name> is not an X.25 trunk.

--ERROR-- Trunk name <trunk_name> may not appear more than once in the list of specified trunks.

--ERROR-- Remote DTE address <remote_dte_address> may not appear more than once in the list of specified remote DTE addresses.

--ERROR-- The remote DTE address <remote_dte_address> contains invalid characters.

--ERROR-- The remote DTE address <remote_dte_address> has an invalid bracketed expression.

--ERROR-- No X.25 incall definition entry for any trunk is defined.

--ERROR-- No X.25 incall definition entry for trunk <trunk_name> is defined.

--FATAL-- Not enough memory available for required table space.

Examples

```
display_x25_incall

X.25 Incall Definition Entries for trunk X25_TRUNK:

  Remote DTE Addresses     123?45[5-7]65*
                           12345
  DTE Address Protocol Id  3 , 8..12
  Accept PDN charges       TRUE

  Remote DTE Addresses     123?54[5-7]65*
                           12345
  DTE Address Protocol Id  6 , 55..66
  Accept PDN charges       FALSE

X.25 Incall Definition Entries for trunk TRANSPAC:

  Remote DTE Addresses     123?45[5-7]65*
                           12345
                           345678901
  DTE Address Protocol Id  3 , 8..12
  Accept PDN charges       FALSE


display_x25_incall trunk_name=ACCUNET

No X.25 incall definition entry for trunk ACCUNET is defined.

display_x25_incall trunk_name=X25_TRUNK remote_dte_addresses=123456

 --ERROR-- An X.25 incall definition entry for the remote DTE address
           123456 is not defined.
```

DISPLAY_X25_INTERFACE_OPTIONS (DISXIO) NETOU Command

Purpose Displays the configuration of an X.25 packet level interface. The packet level interface may provide interface for an X.25 gateway, X.25 AsyncTip or for an X.25 network solution.

Format

DISPLAY_X25_INTERFACE_OPTIONS
INTERFACE_NAME = list 1..15 of name
DISPLAY_OPTION = list of keyword values

Parameters

INTERFACE_NAME (IN)

Specifies the logical name(s) of the packet level interface to be displayed. The name was previously assigned by a DEFINE command. If an INTERFACE_NAME is not specified, the display presents configuration information for all packet level interfaces defined for the system.

DISPLAY_OPTION (DO)

Specifies one or more of the interface attributes for display. These attributes are defined by parameters on the DEFXI command that configured the line. For more information on these parameters, see the DEFXI command description in this manual. The allowed keyword values are:

TRUNK_NAME (TN)
PUBLIC_DATA_NETWORK (PDN)
INTERFACE_NAME (IN)
LOCAL_DTE_ADDRESS (LDA)
PACKET_SEQUENCE_NUMBERING (PSN)
PVC_RANGE (PR)
INONLY_RANGE (IR)
TWOWAY_RANGE (TR)
OUTONLY_RANGE (OR)
DEFAULT_WINDOW_SIZE (DWS)
DEFAULT_PACKET_SIZE (DPS)
CONGESTED_THRESHOLD (CT)
INTERFACE_STANDARD (IS)
SUBSCRIBED_FACILITIES (SF)
ALL (A)

Default is ALL.

Responses The following header precedes the X.25 interface options display (see example).

X.25 Interface Options

Within the options display, the following response is inserted if the INTERFACE_NAME is not defined.

X.25 interface <name> is not defined.

Within the options display, the following response is inserted if no X.25 interfaces are defined in the system.

No X.25 interfaces are defined for this system.

Examples

```
display_x25_interface_options in=telenet_interface

X.25 Interface Options
trunk_name = TELENET_1
public_data_network = TELENET
interface_name = TELENET_INTERACE
local_dte_address = '6121231234'
packet_sequence_numbering = NORMAL
pvc_range = -NONE_DEFINED
inonly_range = 1..24
twoway_range = 25..32
outonly_range = -NONE_DEFINED
default_window_size = 7
default_packet_size = 128
congested_threshold = 40
interface_standard = CCITT_1980
subscribed_facilities = none
```

DISPLAY_X25_INTERFACE_STATUS (DISXIS)
NETOU Command

Purpose

Displays the operating status of one or more X25 interfaces. The display includes the name and status of the X.25 interface, the name and status of the trunk serving the interface, trunk type, and information related to the service access points (SAPs) and connections serviced by the X.25 interface.

The possible trunk types associated with an X.25 interface are:

Trunk Type	Description
LAPB	The interface is communicating with a PDN (DTE-to-DCE or DCE-to-DTE).
Non-LAPB	The interface is communicating with another DTE (DTE-to-DTE) over a non-LAPB connection (over HDLC).
LLC2	The interface is communicating with another DTE (DTE-to-DTE) using the LLC 2 protocol (over Ethernet).

Format

DISPLAY_X25_INTERFACE_STATUS
INTERFACE_NAME = list 1..15 of name
DISPLAY_OPTION = keyword value

Parameters

INTERFACE_NAME (IN)

Specifies the name of the interface or interfaces for which status is to be displayed. Default is all interfaces.

DISPLAY_OPTION (DO)

Displays the level of status response. The allowed keyword values are:

Keyword Value	Description
SUMMARY (S)	Displays general status information.
EXPANDED (E)	Displays the summary information as well as detailed information regarding the virtual circuits serviced by the interface.
EXTENDED (X)	Displays the same information as the EXPANDED keyword except that the CEPID column is replaced by a column entitled VIRTUAL CIRCUIT NAME, which lists the terminal names corresponding to the logical channel numbers.

Default is SUMMARY.

Responses

The following header precedes the status display (see example).

X.25 Interface Status

If errors occur, one of the following responses appears.

X.25 Interface <name> is not defined.

No X.25 interfaces are defined for this system.

Examples

```
display_x25_interface_status cdcnet_lab_if

X.25 Interface Status
interface name = CDCNET_LAB_IF
interface status = active
trunk name = CDCNET_LAB_TRUNK
trunk status = active
number of SAPs = 2
number of virtual circuits = 5
number of virtual circuits congested = 2
trunk type = LAPB

display_x25_interface_status cdcnet_lab_if do=expanded

X.25 Interface Status
interface name = CDCNET_LAB_IF
interface status = active
trunk name = CDCNET_LAB_TRUNK
trunk status = active
number of SAPs = 2
number of virtual circuits = 5
number of virtual circuits congested = 2
trunk type = LAPB

SAP table address = 107B2A(16)
  protocol ids = (0C0(16), 0C1(16), 0C2(16))

  virtual circuits

  logical  type      priority      circuit      output     cepid
  channel                          congestion   packets/
  number                                        bytes

  2095     pvc       batch         no           0/0        109522(16)
  1163     two way   interactive   yes          5/620      106666(16)
  5        in only   batch         yes          7/917      103952(16)
  22       out only  batch         no           0/0        102344(16)
  99       dynamic   batch         no           0/0        108764(16)

SAP table address = 1075AC(16)
  protocol ids = (0C0(16), 0C1(16), 0C2(16))

  virtual circuits

  logical  type      priority      circuit      output     cepid
  channel                          congestion   packets/
  number                                        bytes

  2999     two way   interactive   no           0/0        107654(16)
```

DISPLAY_X25_NET_OPTIONS (DISXNO) NETOU Command

Purpose Displays the current configuration of an X.25 network.

Format

DISPLAY_X25_NET_OPTIONS
NETWORK_NAME = list 1..15 of name
DISPLAY_OPTION = list 1..12 of keyword value

Parameters

NETWORK_NAME (NN)

Displays a list of logical names, of one or more X.25 networks, that were assigned with a DEFINE_X25_NET command (defined earlier in this manual). Default displays all X.25 networks.

DISPLAY_OPTION (DO)

Displays one or more of the X.25 network attributes. These attributes are defined by parameters on the DEFXA command that configured the line. For more information on these parameters, see the DEFXA command description in this manual.The following keyword values are allowed.

TRUNK_NAME (TN)
REMOTE_DTE_ADDRESS (RDA)
NETWORK_ID (NI)
NETWORK_NAME (NN)
COST (C)
RELAY_ALLOWED (RA)
ROUTING_INFO_NETWORK (RIN)
NETWORK_PROTOCOL_ID (NPI)
ACCEPT_PDN_CHARGES (APC)
OUTPUT_QUEUE_LIMIT (OQL)
INCOMING_SERVICE_ACCESS (ISA)
OUTGOING_SERVICE_ACCESS (OSA)
ALL (A)

Default is ALL.

Responses The following header precedes the X.25 network options display (see example).

X.25 Network options

Within the list of options, one of the following responses is inserted if a network name is not defined, is not an X.25 network or if no X.25 networks are defined, respectively.

Network <name> is not defined.

Network <name> is not an X.25 network.

No X.25 networks are defined for this system.

Examples

```
display_x25_net_options nn=x25_tymnet

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
output_queue_limit = 30000
incoming_service_access = all
incoming_service_access titles = none
outgoing_service_access = none
outgoing_service_access titles = none
```

DISPLAY_X25_SWITCH (DISXS) NETOU Command

Purpose Displays previously configured X.25 switch definition entries for one or more X.25 Gateway names and one or more called DTE addresses.

Format

DISPLAY_X25_SWITCH
GATEWAY_NAME = list 1..32 of name or *ALL*
CALLED_DTE_ADDRESSES = list 1..31 of string 1..31 or *OTHER* or *ALL*

Parameters

GATEWAY_NAME (GN)

Specifies a set of X.25 Gateways associated with the X.25 switch definition entries to be displayed. Use ALL to display all X.25 Gateways previously configured with X.25 switch definition entries. The default is ALL.

CALLED_DTE_ADDRESSES (CDA)

Specifies one or more called DTE addresses. To associate the switch definition with multiple DTE addresses, a DTE address can contain wild card attributes as follows:

[D1-D2] Within a bracketed group, a range of digits can be specified. D1 represents the low end of the range; D2 represents the high end. Any value within the range can be used to match a digit in the position specified in the address. For example, 1234[6-7]59 matches two DTE addresses: 1234659 and 1234759.

[...] Within a bracketed group, any list of single digits can be specified. For example, 1234[179]59 matches three DTE addresses: 1234159, 1234759, and 1234959.

* Any single * can be used to match a string of digits, including the null string. For example, 1234* matches 1234, 12341, 1234562345, and so forth.

? Any single ? can be used to match a digit in the position specified in the address. For example, 1234?59 matches 10 DTE addresses: 1234059, 1234159, 1234259 ... 1234959.

If multiple ADD_X25_SWITCH definitions exist that match a given DTE address, the call definition associated with the most specific string is displayed. For example, 123?56 is more specific than 123* in matching called DTE address 123456.

The value OTHER corresponds to any entries defined with the value OTHER. The value ALL displays all X.25 switch definition entries for the X.25 Gateways specified. The default is ALL.

Responses The configuration for each X.25 Gateway is displayed separately. For each called DTE address for which an X.25 switch is defined, the configurable parameters are displayed. All entries with the same parameter settings are grouped. For the format of this display, see the example.

The following response is generated if no X.25 switch is defined, regardless of input errors:

No X.25 switch definition entry for any X.25 Gateway defined.

The following response is generated if an X.25 switch is defined, but none explicitly for an X.25 Gateway:

No X.25 switch definition entry for X.25 Gateway <gateway_name> is defined.

The following error responses are generated upon an error condition:

--ERROR-- An X.25 switch definition entry for the called DTE address <called_dte_address> is not defined for X.25 Gateway <gateway_name>.

--ERROR-- X.25 Gateway <gateway_name> is not defined.

--ERROR-- X.25 Gateway <gateway_name> may not appear more than once in the list of specified X.25 Gateways.

--ERROR-- Called DTE address <called_dte_address> may not appear more than once in the list of specified called DTE addresses.

--ERROR-- The called DTE address <called_dte_address> contains invalid characters.

--ERROR-- The called DTE address <called_dte_address> has an invalid bracketed expression.

--FATAL-- Not enough memory available for required table space.

Examples

```
display_x25_switch

X.25 Switch Definition Entries for X.25 Gateway X25_GW_SWITCH

  Called DTE Addresses     123?45[5-7]65*
                           12345
  Peer VE Outcall Title    OUTBOUND_SWITCH

  Called DTE Addresses     987?45[5-7]65*
  Peer VE Outcall Title    OUTBOUND_SWITCH_2

X.25 Switch Definition Entries for X.25 Gateway X25_SWITCH

  Called DTE Addresses     123?45[5-7]65*
                           98765
  Peer VE Outcall Title    OUTBOUND_SWITCH


display_x25_switch PEER_X25_GW

No X.25 switch definition entry for X.25 Gateway PEER_X25_GW is
defined.


display_x25_switch gateway_name=PEER_X25_GW called_dte_addresses='123456'

--ERROR-- An X.25 switch definition entry for the called DTE address
123456 is not defined for X.25 Gateway PEER_X25_GW.
```

DISPLAY_X25_TERMINAL_GW (DISXTG) NETOU Command

Purpose Displays the configuration of an X.25 Terminal Gateway. The display returns the CDCNET title, by which the gateway is known in the catenet, and other pertinent configuration information.

Format

DISPLAY_X25_TERMINAL_GW
GATEWAY_NAME = list 1..15 of name
DISPLAY_OPTION = list of keyword value

Parameters *GATEWAY_NAME (GN)*

Specifies the logical name of one or more X.25 terminal gateway definitions. Default displays the configuration of all X.25 terminal gateways defined for the system.

DISPLAY_OPTION (DO)

Specifies one or more of the gateway attributes for display. These attributes are defined by parameters on the DEFXT command that configured the line. For more information on these parameters, see the DEFXT command description in this manual. The following keyword values are allowed:

TRUNK_NAME (TN)
TITLE (T)
OUTCALL_DEFINITION_REQUIRED (ODR)
ALL (A)

Default is ALL.

Responses The following header precedes the options display (see example).

X.25 terminal gateway options

If errors occur, one of the following responses appears.

X.25 terminal gateway <name> is not defined.

No X.25 terminal gateway is defined.

Examples

```
display_x25_terminal_gw

X.25 terminal gateway options
gateway_name = DATAPAC_TERMINAL_GATEWAY
trunk_name = (DATAPAC_TRUNK)
title = DATAPAC_ACCESS
outcall_definition_required = FALSE
```

DISPLAY_X25_TERMINAL_GW_OUTCALL (DISXTGO) NETOU Command

Purpose Displays the outcall configuration of an X.25 terminal gateway. The display returns the outcall information associated with the gateway name.

Format

DISPLAY_X25_TERMINAL_GW_OUTCALL
GATEWAY_NAME = list 1..15 of name or *ALL*
DISPLAY_OPTION = list of keyword value

Parameters

GATEWAY_NAME (GN)

Displays the names of the X.25 terminal gateways which provide access to remote applications. Defaults to ALL.

DISPLAY_OPTION (DO)

Displays one or more of the outcall attributes. These attributes are defined by parameters on the DEFXTG command that configured the line. For more information on these parameters, see the DEFXTG command description in this manual. The allowed keyword values are:

OUTCALL_NAME (ON)
REMOTE_DTE_ADDRESS (RDA)
LOCAL_DTE_ADDRESS (LDA)
FACILITIES (F)
USER_DATA (UD)
ALL (A)

Default is ALL.

Responses

--ERROR-- X.25 terminal gateway <name> is not defined.

--ERROR-- No X.25 terminal gateway is defined.

Examples

```
display_x25_terminal_gw_outcall

X.25 terminal gateway outcall options
gateway_name = TELENET_TERMINAL_GATEWAY

  outcall_name = SOLVER1
  remote_dte_address = 311040800146
  local_dte_address = none
  facilities = none
  user_data = none

  outcall_name = RANGE_408
  remote_dte_access = 3110408*

  local_dte_address = none
  facilities = none
  user_data = none
```

DISPLAY_X25_TRUNK_OPTIONS (DISXTO) NETOU Command

Purpose Displays the configuration of X.25 trunks. An X.25 trunk is addressed by its trunk name. If a trunk name is not entered, the configuration of all X.25 trunks defined for the system is displayed .

Format

DISPLAY_X25_TRUNK_OPTIONS
TRUNK_NAME = list 1..15 of name
DISPLAY_OPTION = list of keyword value

Parameters

TRUNK_NAME (TN)

Specifies the logical name of an X.25 trunk assigned via a DEFINE_X25_TRUNK command.

DISPLAY_OPTION (DO)

Specifies one or more of the trunk attributes for display. The following keyword values are allowed:

LIM (L)
PORT (P)
TRUNK_NAME (TN)
MODE (M)
MAX_UNACK_FRAMES (MUF)
PF_RECOVERY_TIMER (PRT)
ERROR_RECOVERY_TIMER (ERT)
RETRANSMISSION_LIMIT (RL)
TRUNK_SPEED (TS)
CLOCKING (C)
INTERACTIVE_BANDWIDTH (IB)
DRIVER (D)
MAX_FRAME_SIZE (MFS)
ALL (A)

Default is ALL.

Responses The following header precedes the options status display (see example).

X.25 Trunk options

If errors occur, one of the following responses appears.

Trunk <name> is not defined.

Trunk <name> is not an X.25 trunk.

No X.25 trunks are defined for this system.

Examples

```
display_x25_trunk_options tn=x25_telenet_1

X.25 Trunk options
lim = 3
port = 1
trunk_name = X24_TELENET_1
mode = DTE
max_unack_frames = 7
pf_recovery_timer =  500
retransmission_limit = 20
trunk_speed = 19200
clocking = internal
interactive_bandwidth = 7
driver = hdlc_normal
max_frame_size = 71500
```

EXEET, HELP, IDLSD, IDLTS, and KILS Commands 9

EXEET, HELP, IDLSD, IDLTS, and KILS Commands 9

This chapter provides complete descriptions of CDCNET commands beginning with the letter E through K. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

EXECUTE_ECHO_TEST (EXEET)
NETOU Command

Purpose Executes the inline diagnostic path test between any two CDCNET OSI systems. This test does not involve sending any data.

There is no STOP command for this test. Instead, a response is issued by the command processor indicating the test status. The response includes the destination system name tested.

Format

EXECUTE_ECHO_TEST
DESTINATION_SYSTEM = name

Parameters

DESTINATION_SYSTEM (DS)

Specifies the system name of the OSI systems to be verified.

Responses Echo Test to <destination_system> PASSED. Response time (msec.) = <integer>

--ERROR--System <destination_system> is unknown.

--ERROR--Source system <system_name> is NOT a CDCNET OSI system.

--ERROR--Destination system <system_name> is not a CDCNET OSI system.

--ERROR--Echo Test to <destination_system> FAILED. < one line of additional error message >

"Unsuccessful CLNS Data Request."

"No response from the Destination System."

"Destination System is NOT a CDCNET OSI system."

"No Echo Test Task ID Table Space Available."

"System Error - No data buffers available."

Remarks The only restriction on the path used to to reach the destination system is that all systems in the path must be OSI systems. The path selection is done by CLNS software.

Examples The following example verifies that a path exists between the two DIs on system name tdi_e3.

```
exeet ds=tdi_e3

 Echo Test to TDI_E3 PASSED
 Response time (msec) = 3420
```

HELP
NETOU Command

Purpose Performs the same function as the DISPLAY_COMMAND_LIST command. Refer to the DISPLAY_COMMAND_LIST command in this manual.

Format **HELP**

Parameters None.

Responses Alphabetical list of all network commands (see example).

Examples

```
help

add_np_gw_outcall                    add_x25_gw_outcall
        .                                    .
        .                                    .
        .                                    .
unload_module                        write_terminal_message
```

IDLE_SERVER_DEVICES (IDLSD)
NETOU Command

Purpose Specifies the servers which are to be gradually idled down.

Format

IDLE_SERVER_DEVICES
TITLES = list of name or *ALL*
IDLE_MESSAGE = list 1..4 of string

Parameters

TITLES (T)

Specifies the title or titles of the servers for which access is to be gradually idled down. Default is ALL, meaning that access to all servers is to be idled down.

IDLE_MESSAGE (IM)

Specifies the text string to be sent to users that attempt to access the idled servers specified by the titles parameter. Default is the empty string.

Responses Server Devices are idled.

--WARNING--No Server Devices were found.

--WARNING-- Server Devices <title> already idled.

--ERROR-- Server Devices <title> not defined.

Examples

```
idle_server_devices

Server Devices are idled.
```

IDLE_TELNET_SERVER (IDLTS) NETOU Command

Purpose Performs a gradual shutdown of access to one or more TELNET servers. Users currently connected to servers at the time this command is issued, are allowed to complete their existing sessions. New connections are not allowed until a corresponding RESUME_TELNET_SERVER command is executed.

Format **IDLE_TELNET_SERVER**
SERVER_NAME = list of name(s) or *ALL*

Parameters *SERVER_NAME (SN)*

Specifies the names of the servers for which access is to be gradually idled down. Default is ALL.

Responses TELNET server <server_name> is idled.

--WARNING-- No TELNET servers were found.

--WARNING-- TELNET server <server_name> is already idled.

--WARNING-- TELNET server <server_name> was not found.

Examples

```
idle_telnet_servers (ARH1, ARH2, ARH3)

TELNET server ARH1 is idled.
--WARNING-- TELNET server ARH2 is already idled.
--WARNING-- TELNET server ARH3 was not found.


idle_telnet_servers

--WARNING-- No TELNET servers were found.
```

KILL_SYSTEM (KILS)
NETOU Command

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 are going to be disconnected from CDCNET services by sending them a message using the WRITE_TERMINAL_MESSAGE command described in this manual.

Format

KILL_SYSTEM
DUMP = boolean
COMPRESS_DUMP = boolean

Parameters *DUMP (D)*

Requests a full DI memory dump before reload. Possible parameter values are YES and NO. Default is NO.

COMPRESS_DUMP (CD)

Requests a compressed dump (omits free memory and buffers) of DI memory. Possible parameter values are YES and NO. Default is YES.

Responses System being reset and reloaded.

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 see the CDCNET Configuration Guide.

Examples

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

System being reset and reloaded.
```

LOAF, LOAM, RESSD, and, RESTS Commands 10

LOAF, LOAM, RESSD, and, RESTS Commands 10

This chapter provides complete descriptions of the CDCNET Network Operations LOAD and RESUME commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

LOAD_FILE (LOAF)
DI Configuration/NETOU Command

Purpose

Makes a CDCNET file DI-memory-resident. The file types which can be loaded are EXCEPTION_LIST, TERMINAL_DEFINITION_PROCEDURE, TERMINAL_USER_PROCEDURE, DEVICE_LOAD_PROCEDURE, and VALIDATION.

Changing file residence affects only file references performed by the DI where LOAD_FILE is executed. Making a file DI resident allows faster access and does not require a CYBER host once the file is loaded. Thus, certain files, such as TUPs needed for the Interactive Passthrough Gateway, can be accessed even if there is no active CYBER host available.

Since the file is stored in a buffer chain, the memory used is more than the actual file size (there is a 46-byte overhead for each data buffer allocated by the system). If you use abbreviated command and parameter names, text files can be stored using less memory.

If the memory required to load the file is unavailable, an error message is issued. When this happens while loading a new version of a file already residing in DI memory, the old version remains in DI memory and is used to satisfy subsequent file requests. If you don't want this to happen, you can unload the old file using UNLOAD_FILE. Once the file is unloaded, subsequent file requests are satisfied from a file version not residing in the DI.

NOTE

Control Data strongly recommends using LOAD_FILE to load a TUP for a passthrough port into DI memory. Loading the TUP into DI memory ensures that the TUP, which must be frequently accessed and executed for a passthrough configuration, is available at all times. Once the TUP is loaded into DI memory, the TUP's availability is not subject to a CYBER host's availability.

Format

LOAD_FILE
FILE = name or string 1..63
FILE_TYPE = keyword value
MAXIMUM_FILE_SIZE = integer 1..65000

Parameters

FILE (F)

The name of the file to be loaded. If the named file already resides in the DI, it is reloaded with a new, possibly different, copy.

File names entered as strings are converted to all uppercase and are restricted to the character set of SCL type name.

FILE_TYPE (FT)

A file type. Valid file types include the following:

Keyword Value	Description
EXCEPTION_LIST, EL, EXCEPTION	Exception list
TERMINAL_DEFINITION_PROCEDURE, TDP, TERMINAL_PROCEDURE	Terminal definition procedure (TDP)
TERMINAL_USER_PROCEDURE, TUP, USER_PROCEDURE	Terminal user procedure (TUP)
DEVICE_LOAD_PROCEDURE, DLP, LOAD_PROCEDURE	Device load procedure
VALIDATION	User validation

The default is TERMINAL_USER_PROCEDURE.

NOTE

The format of VALIDATION file names is 'username@domain'. Both username and domain are 1..31 characters and must be specified.

MAXIMUM_FILE_SIZE (MFS)

The maximum number of bytes of DI memory that the requested file can consume (see preceding information under Purpose). A file larger than this value cannot be loaded into a DI. Default is 2500 bytes.

Responses

File <name> is loaded, size is <nnnnnn> bytes.

--ERROR-- Maximum size for file <name> is exceeded, size is <nnnnnn> bytes.

--ERROR-- Unable to locate file <name> with file type <type>.

--ERROR-- Unable to load file <name>, <file access error template> [file source system-id is <system-id>].

--FATAL-- Not enough system buffers are currently available to load file <name>, size is <nnnnnn> bytes.

Examples

```
load_file f=printer_with_vfu_p31310 ft=terminal_procedure

File printer_with_vfu_p31310 is loaded from system1, size
is 2500 bytes.
```

LOAD_MODULE (LOAM)
DI Configuration/NETOU Command

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 sets only the retain status for the module; it does not guarantee that a new copy of the module is loaded. A retained module is not unloaded to recover system memory resources, even if the module is unused and memory resources are scarce.

Format

LOAD_MODULE
MODULE = name
ENTRY_POINT = name
RETAIN = boolean

Parameters *MODULE (M)*

The name of the desired software module.

ENTRY_POINT (EP)

The entry point name used to determine which module to load from the DI library.

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-- Entry point <entry point> is not contained within module <module>.

--ERROR-- Module <module> was not found in directory.

--ERROR-- Module for entry point <entry point> was not found.

--ERROR-- One of the parameters Module or Entry Point must be specified.

--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-- Not enough buffers are currently available to load module <module>.

--FATAL-- Not enough buffers or memory are 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-- 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>.

--FATAL-- Unrecognizable record in module <module>.

--FATAL-- Unknown record kind encountered in module <module>.

Remarks To display the modules currently loaded in a DI, send the DISPLAY_SOFTWARE_LOAD_STATUS (see description in this manual) command to the DI. An alternative method is to use the NPA report LOADRP1 to identify modules loaded per DI. See the Network Performance Analyzer commands in this manual for the procedure used to generate 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. Then 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.

```
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.

```
load_module m=display_hardware_status r=no

Module DISPLAY_HARDWARE_STATUS loaded.
```

RESUME_SERVER_DEVICES (RESSD)
NETOU Command

Purpose Resumes access to Passthrough and/or Device Outcall Servers that have had their access previously idled down.

Format

RESUME_SERVER_DEVICES
TITLES = list of name or *ALL*

Parameters *TITLES (T)*

Specifies the title or titles of the servers for which access is to be resumed. Default is ALL, meaning that access to all servers is to be resumed.

Responses Server Devices are resumed.

--WARNING-- Server Devices <title> not idled.

--ERROR-- Server Devices <title> not defined.

Examples

```
resume_server_devices

Server Devices are resumed.
```

RESUME_TELNET_SERVER (RESTS)
NETOU Command

Purpose Resumes access to TELNET servers that have had their access previously idled down.

Format

RESUME_TELNET_SERVER
SERVER_NAME = list of name(s) or *ALL*

Parameters *SERVER_NAME (SN)*

Specifies the names of the servers for which access is to be resumed. Default is ALL.

Responses TELNET server <server_name> is resumed.

--WARNING-- No TELNET servers were found.

--WARNING-- TELNET server <server_name> is not idled.

--WARNING-- TELNET server <server_name> was not found.

Examples

```
resume_telnet_servers (ARH1, ARH2, ARH3)

TELNET server ARH1 is resumed.
--WARNING-- TELNET server ARH2 is not idled.
--WARNING-- TELNET server ARH3 was not found.

resume_telnet_servers

--WARNING-- No TELNET servers were found.
```

SENDD and SETDAT Commands 11

SENDD and SETDAT Commands 11

This chapter provides complete descriptions of the following CDCNET commands:

SEND_DIAGNOSTIC_DATA (SENDD)
SET_DATE_AND_TIME (SETDAT)

Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

SEND_DIAGNOSTIC_DATA (SENDD) NETOU Command

Purpose Sends a diagnostic data string to a LIM port. Use this command to send commands to an intelligent modem on an asynchronous line. Place the LIM port in the DOWN state before executing this command.

Format

SEND_DIAGNOSTIC_DATA
DEVICE_NAME = name
DATA = string of 1..64
CARRIAGE_RETURN = boolean
RESPONSE_TIMEOUT = integer 1..10

Parameters

DEVICE_NAME (DN)

Specifies the physical name of the device. The name consists of the keyword LIM, the LIM board slot number, the keyword PORT, and the port number on the LIM.

DATA (D)

Specfies the desired diagnostic data string to be sent to the LIM port. The data string must be enclosed in two single quotes. Thus, when the data string ends the command, two single quotes end the data string and a third single quote ends the entire command. See the Examples below.

CARRIAGE_RETURN (CR)

Specifies a flag indicating whether a carriage return should be appended to the diagnostic data string. Default is TRUE.

RESPONSE_TIMEOUT (RT)

Specifies the time in seconds to wait for the intelligent modem to return its response. Default is 1 second.

Responses Response data follows: <response data>

--WARNING-- Device <device_name> test already started.

--WARNING-- Response data was not received.

--ERROR-- Expecting device name $LIMx_PORTx, found <device_name>

--ERROR-- Device <device_name> not installed in system

--ERROR-- Device <device_name> not in "DOWN" state.

--ERROR-- Configured line is not ASYNC.

--FATAL-- Aborted trying to start up line.

Examples

```
send_diagnostic_data device_name=$LIM4_PORT0 D=''ATU0''

Response data follows:
```

The following example illustrates how to use the SEND_DIAGNOSTIC_DATA command to test an intelligent modem without using the existing online port diagnostic.

Place the port in the DOWN state, with the CHANGE_ELEMENT_STATE command.

```
change_element_state device_name=$LIM0_PORT0 ..
s=down
```

The following response appears:

```
Device $lim0_port0 down
```

Place the modem in local analog loopback mode with the SEND_DIAGNOSTIC_DATA command.

```
send diagnostic_data dn=$LIM0_PORT0 ..
data=''ATU0'' RT=1
```

The following response appears:

```
Response data follows:

ATU0
```

Send a data string to the modem, with the SEND_DIAGNOSTIC_DATA command. The data should loop back to the modem.

```
send_diagnostic_data device_name=$LIM0_PORT0 ..
data=''UUUUUUUUUUUUUUUUUUUU''
```

The following response appears:

```
Response data follows:

UUUUUUUUUUUUUUUUUUUU
```

Terminate the modem local analog loopback mode with the SEND_DIAGNOSTIC_DATA command.

```
send_diagnostic_data device_name=$LIM0_PORT0 ..
data=''+++''carriage_return=FALSE
```

The following response appears:

```
Response data follows:

+++
```

Return the port to the ON state with the CHANGE_ELEMENT_STATE command.

```
change_element_state device_name=$LIM0_PORT0 ..
state=ON
```

The following response appears:

```
Device $lim0_port0 on
```

The following example illustrates how to use the SEND_DIAGNOSTIC_DATA command to test an intelligent modem with the existing online port diagnostic in modem loopback mode.

Put the port in the DOWN state with the CHANGE_ELEMENT_STATE command.

```
change_element_state device_name=$LIM0_PORT0 ..
state=DOWN
```

The following response appears:

```
Device $lim0_port0 down
```

Place the modem in local analog loopback mode with the SEND_DIAGNOSTIC_DATA command.

```
send_diagnostic_data device_name=$LIM0_PORT0 ..
data=''ATU0''
```

The following response appears:

```
Response data follows:

ATU0
```

Execute the port online diagnostic in modem loopback mode with the START_PORT_TEST command.

```
start_port_test device_name=$LIM0_PORT0 ..
loop_mode=MODEM success_state=DOWN modem_class=4
```

NOTE

The example assumes the test passes. SUCCESS_STATE=DOWN is required to eliminate the need for another CHAES command to place the port in the DOWN state. MODEM_CLASS=4 is needed if the line (port) is undefined or defined as an auto_rec line.

Terminate modem local analog loopback mode and return port to the ON state as in previous example.

SET_DATE_AND_TIME (SETDAT) NETOU Command

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 are set, you can send the SYNCHRONIZE_CLOCK command to each DI in the network (see SYNCHRONIZE_CLOCK command description in this manual), 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 depends 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 is 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 is 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 The following header precedes the date and time in selected format (see example).

Master clock for catenet set.

If an error occurs, one of the following headers appears.

--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 described in this manual. To determine which DI is configured to be the clocking_system, send the DISPLAY_SYSTEM_OPTIONS (DISSO) command, described in this manual, 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 changes to 1986, but the current day and month do not change. 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

```
set_date_and_time d=''24/11/85'',..
t=''08:25:49''

Master clock for catenet set
24/11/85 08:25:49
```

START Commands 12

This chapter provides complete descriptions of all CDCNET Network Operations START commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

START_CIM_TEST (STACT)
NETOU Command

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 later in this chapter), and if no errors occur while running the LIM test, the Port test should be run (see START_PORT_TEST command later in this chapter).

Format

START_CIM_TEST
DEVICE_NAME = keyword value
REPEAT_PASS = integer 0..65535
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 value 0 specifies that the test is run continuously until you stop the test by a STOP_CIM_TEST command described in this manual. Default is 1.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error causes 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 is left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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 The following message is a success response.

CIM test started, version <version_number>. CIM slot number = <cim slot number>.

If an error occurs, one of the following messages is received.

--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, described in this manual, to change the state of the device.

To get the results of the CIM test, send the DISPLAY_TEST_STATUS command, described in this manual, 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 (described in this chapter). However, you should still run the port test (using START_PORT_TEST described in this chapter), 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 described in this manual. 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.

```
start_cim_test device_name=$cim5

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 is run continuously without stopping for errors. However, since logging is on, any errors encountered during the test are logged.

```
start_cim_test dn=$cim5,rp=0,soe=off

CIM test started, version 0901
CIM slot number = 5
```

START_EGP_PEER (STAEP)
NETOU Command

Purpose Starts Exterior Gateway Protocol (EGP) processing for the specified peer. If the peer has not already been acquired and the peer was defined as AUTO or MANUAL mode (see the PEER_USAGE parameter of the DEFINE_EGP_PEER command described in this manual), then EGP attempts to acquire the peer and exchange routing information. If the peer has not been started and the peer was defined for BACKUP mode, the peer is eligible for backup duty if an active peer is lost. If the peer has already been started and acquired, EGP generates a 'start' event that forces EGP to reacquire the peer.

Format

START_EGP_PEER
IP_ADDRESS = list 4 of (0..255)

Parameters

IP_ADDRESS (IA)

Identifies the IP address of the EGP peer gateway with which exchange of routing information is to be started.

Responses

EGP peer <ip_address> is started.

EGP peer <ip_address> enabled for BACKUP use.

EGP peer <ip_address> is already acquired - start event generated.

--ERROR-- EGP peer <ip_address> is not defined.

Examples

```
start_egp_peer ip_address = (128,5,0,3)

EGP peer 128.5.0.3 is started.
```

START_ESCI_TEST (STAET)
NETOU Command

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 described in this manual; since communications must be stopped on the device being tested before the diagnostics can be executed.

Format

START_ESCI_TEST
DEVICE_NAME = keyword value
REPEAT_PASS = integer 0..65535
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 value 0 specifies that the test is run continuously until you stop the test by a STOP_ESCI_TEST command described in this manual. Default is 1.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error causes 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 is left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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

The following message is a success response.

ESCI test started, version <version_number>.
ESCI slot number = <esci slot number tested>.

If an error occurs, one of the following messages is received.

--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 described in this manual to change the state of the device.

To get the results of the ESCI test, send the DISPLAY_TEST_STATUS command described in this manual to the DI that contains the device being tested.

If you specify SUCCESS_STATE=DOWN, you must use the CHANGE_ELEMENT_STATE command described in this manual, 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 are logged.

```
start_esci_test device_name=$esci6,l=off

ESCI test started, version 0901
ESCI slot number = 6
```

START_LIM_TEST (STALT)
NETOU Command

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 described in this chapter). If problems are reported on more than one LIM, the CIM diagnostic test should be run (see START_CIM_TEST described in this chapter).

Format

START_LIM_TEST
DEVICE_NAME = keyword value
REPEAT_PASS = integer 0..65535
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 value 0 specifies that the test is run continuously until you stop the test by a STOP_LIM_TEST command described in this manual. Default is 1.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error causes 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 is left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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 You receive the following response if the test starts successfully.

```
LIM test started, version <version_number>.
CIM slot number= <CIM slot number>.
LIM slot number= <lim slot number>.
```

The following response indicates the test has already been started.

```
--WARNING-- Device <device_name> test already started.
```

The following responses indicate the device cannot be tested.

```
--ERROR-- Device <device_name> not installed in system.
```

```
--ERROR-- Device <device_name> not in "DOWN" state.
```

```
--ERROR-- Device <device_name> cannot be tested at this time.
$CIM <cim slot number> must be dedicated to the <driver_type>
driver and is currently active. The STALT command may be reissued
once all active lines/trunks on $CIM <cim slot number> are stopped.
```

```
--FATAL-- LIM test aborted, version <version_number>.
CIM slot number= <cim slot number>.
Unable to start test task.
```

You receive the following response when the test task started but terminated prematurely.

```
--FATAL-- LIM test aborted, version <version_number>.
CIM slot number= <cim slot number>.
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_ELEMENT_STATE command to change the hardware to the appropriate state. See the CHANGE_ELEMENT_STATE command described in this manual.

```
--FATAL-- LIM test aborted, version <version_number>.
CIM slot number= <cim slot number>.
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 described in this manual to determine the status of on-board tests.

```
--FATAL-- LIM test aborted, version <version_number>.
CIM slot number= <cim slot number>.
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 can not start a LIM test until you run a CIM test (using the START_CIM_TEST command described in this manual).

--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 described in this manual to change the state of the device.

To get the results of the LIM test, send the DISPLAY_TEST_STATUS command, described in this manual, to the DI that contains the device being tested.

If you specify SUCCESS_STATE=DOWN, you must use the CHANGE_ELEMENT_STATE command described in this manual when the diagnostic completes to put the device in the ON state.

Examples

```
start_lim_test device_name=$lim5

LIM test started, version 10H3.
CIM slot number = 6.
LIM slot number = 5.
```

START_LINE (STAL)
NETOU Command

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

```
start_line line_name=line31

Line LINE31 started.
```

START_LINE _METRICS (STALM)
NETOU Command

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 are reported. The maximum interval is 24 hours (86,400 seconds).

GROUP (G)

Specifies the type of statistics group requested to be collected:

SUMMARY (S)
EXPANDED (E)
ALL (A)

Default is SUMMARY statistics.

REPORT (R)

Specifies whether or not statistics should be reported by log messages. The messages are generated and sent to the CDCNET log file according to the interval set by the REPORT_INTERVAL command described in this manual. Possible values are YES, generate reporting log messages, and NO, do not generate reporting log messages. Default is YES.

Responses The following response indicates the test successfully started (one response for each line and group for which metrics was started).

Line <line_name> <group_name> metrics 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. When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

Reporting log message <message_number> not enabled.

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 described in this manual. If it is not enabled, enable it using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

See the NPA commands in this manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
start_line_metrics ln=bld_3_async_22,g=all,ri=360

Line BLD_3_ASYNC_22 summary metrics started.
Line BLD_3_ASYNC_22 expanded metrics started.
```

START_MCI_INLINE_TEST (STAMIT) NETOU Command

Purpose Starts the inline diagnostics testing of an MCI board. An inline diagnostics test shares access to the device being tested with nondiagnostic

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 = 1..10000
MESSAGE_LENGTH = keyword value
MESSAGE_INTERVAL = 0..60000

Parameters

DEVICE_NAME (DN)

Specifies the physical name of the MCI to be tested.

MESSAGE_COUNT (MC)

Specifies the number of messages to be transmitted and received as part of this inline test. Default is 100.

MESSAGE_LENGTH (ML)

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

Default 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. Default is 0.

Responses The following message is a success response.

MCI in line test, version <version>
started for device <device_name>

If an error occurs, one of the following messages is received.

--WARNING-- In line 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, VE interface, or channel network solution for device <device_name> is not defined).

--ERROR-- NP interface for device <device_name> is not up.

--ERROR-- VE interface for device <device_name> is not up.

--ERROR-- Unable to start the MCI in line test.
Not enough memory is available for the required table space.

--ERROR-- Unable to start the MCI inline diagnostics task.

Examples

```
start_mci_inline_test device_name = $mci7

MCI in line test, version 2605
started for device $mci7
```

START_MCI_TEST (STAMT)
NETOU Command

Purpose Starts the online diagnostic test on an MCI board.

NOTE

STAMT does not work on NOS/VE channels.

Format

START_MCI_TEST
DEVICE_NAME = keyword value
REPEAT_PASS = integer 0..65535
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 value 0 specifies that the test is run continuously until you stop the test by a STOP_MCI_TEST command described in this manual. Default is 1.

SUCCESS_STATE (SS)

Determines the state in which the hardware device is left in upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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.

Responses The following message is a success response.

MCI test started, version <version number>.
MCI slot number= <mci slot number>.

If an error occurs, one of the following messages is received.

--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

```
start_mci_test device_name=$mci7

MCI test started, version 10H3
MCI slot number= 7
```

START_NET_DELAY_MEASUREMENT (STANDM) NETOU Command

Purpose Starts the network delay measurement. The start processor validates the input parameters, loads and starts the measurement task, and issues a command response indicating the start status of that task.

Format

START_NET_DELAY_MEASUREMENT
DESTINATION_SYSTEM = name
DELAY_TIME_THRESHOLD = integer 0..7FFFFFFF
AVERAGE_ONLY = boolean
MEASUREMENTS = integer 1..7FFFFFFF or *CONTINUOUS*
MESSAGES_PER_MEASUREMENT = integer 1..7FFFFFFF
MESSAGE_PRIORITY = keyword
INTER_MEASUREMENT_INTERVAL = integer 0..7FFFFFFF
ERROR_THRESHOLD = 1..messages_per_measurement
MESSAGE_TIMEOUT = integer 1..7FFFFFFF

Parameters

DESTINATION_SYSTEM (DS)

Specifies the system name which the network delay measurement is to be made from the source (originating) system. This is the name as designated by the SYSTEM_NAME parameter on the DEFINE_SYSTEM command.

DELAY_TIME_THRESHOLD (DTT)

Specifies the delay time threshold in milliseconds. The threshold is compared against the calculated one-way message delay time. This parameter is required if AVERAGE_ONLY is NO.

There is no default.

AVERAGE_ONLY (AO)

Specifies a mode selection allowing the average delay time to be calculated and reported but no delay time threshold comparisons made.

Default is NO.

MEASUREMENTS (M)

Specifies the number of times a measurement is made or repeated. A value of 0 is set to a value of 1. The keyword CONTINUOUS (C) allows the measurement to be made until a stop command is issued.

Default is CONTINUOUS.

MESSAGES_PER_MEASUREMENT (MPM)

Specifies the number of consecutive messages sent within a measurement.

Default is 25.

MESSAGE_PRIORITY (MP)

Specifies the priority at which messages are sent. Allowed values are:

INTERACTIVE (I)
BATCH (B)
SYSTEM (S)
REAL_TIME (R)

Default is INTERACTIVE.

INTER_MEASUREMENT_INTERVAL (IMI)

Specifies the time interval in minutes between transmission of groups of consecutive messages. A value of 0 causes the measurement to run until the measurements count is reached.

Default is 60 (1 hour).

ERROR_THRESHOLD (ET)

Specifies the number of messages exceeding the DELAY_TIME_THRESHOLD in a measurement before the condition is reported.

Default is 1.

MESSAGE_TIMEOUT (MT)

Specifies the amount of time, in seconds allowed for a message to be returned from a destination system. A value of 0 causes the measurement to terminate immediately.

Default is 120 (2 minutes).

Responses

Network Delay Measurement STARTED to <destination_system>.

--WARNING-- "inter_measurement_interval" is zero.

--WARNING-- "delay_time_threshold" is zero.

--WARNING-- "message_timeout" is less than "delay_time_threshold".

--ERROR-- "delay_time_threshold" not specified.

--ERROR-- "error_threshold" cannot be greater than "message_per_measurement".

--ERROR-- Network Delay Measurement to <destination_system> is already active.

--ERROR-- System <destination_system> is unknown.

--ERROR-- Destination system <destination_system> is not a CDCNET OSI system.

--FATAL-- Command aborted. Unable to start measurement task.

Examples

```
senc 'start_net_delay_measurement destination_system=ndi_d2..
delay_time_threshold=400' s=mdi_a1

FROM MDI_A1
 Network Delay Measurement STARTED to NDI_D2.
```

START_NETWORK (STAN)
NETOU Command

Purpose Starts communications over an X.25, Ethernet, or HDLC network solution. Also starts the underlying X.25, Ethernet, or HDLC trunk. For HDLC rotary (multiple trunk) networks, all trunks that are started are shown.

A START_NETWORK command is the same as a START_TRUNK command except for HDLC rotary networks. Both START_NETWORK and START_TRUNK start the communication media. A START_TRUNK command, however, does not work for X.25 trunks; only the START_NETWORK or START_X25_INTERFACE command starts the communication media for X.25 trunks. The only time it makes sense to use START_TRUNK is for HDLC rotaries. With HDLC rotaries, more than one trunk is associated with the specified HDLC network. To start only one rotary trunk, use START_TRUNK. To start all trunks, use START_NETWORK.

Format

```
START_NETWORK
   NETWORK_NAME = name
```

Parameters **NETWORK_NAME (NN)**

The logical name of the network assigned by the define command that configured the network solution.

Responses <Network_type> network <name> started for trunk <trunk_name>.

<Network_type> network <name> started for trunks (<trunk_name> <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-- Unable to start task <entry_point_name>.

--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.

--FATAL-- Unable to start task <entry_point_name>.

Examples

```
start_network network_name=plymouth_net_1

ETHERNET Network PLYMOUTH_NET_1 started for trunk
PLYMOUTH_TRUNK_1.
```

START_NETWORK_METRICS (STANM) NETOU Command

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 pseudo network solutions).

Format

```
START_NETWORK_METRICS
    NETWORK_NAME = list 1..15 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 or the interface name for a NOS/VE channel (defined by the DEFVI command). 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 described in this manual. The network name for a NOS/VE mainframe channel interface is defined by the INTERFACE_NAME parameter on the DEFVI command described in this manual.

For this CDCNET release, the network name of a NOS mainframe channel interface 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 described in this manual.

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 (S)
EXPANDED (E)
ALL (A)

Default is SUMMARY.

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

The following response indicates the network metrics have been successfully started.

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. When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

Reporting log message <message_number> not enabled.

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.

The following error messages may be received.

--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 Ethernet statistics to be reported, log message number 639 must be enabled. To check if this message is enabled, use the DISPLAY_SOURCE_LOG_GROUP command described in this manual. If this message is not enabled, enable it using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

See the NPA commands in this manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
start_network_metrics nn=bld_3_ethernet,ri=360,g=(summary,expanded)

Network BLD_3_ETHERNET summary metrics started.
Network BLD_3_ETHERNET expanded metrics started.
```

START_NP_INTERFACE (STANI) (NOS MDI Only) NETOU Command

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 described in this manual.

Responses

NP_interface <interface_name> started.

--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.

--FATAL-- NP interface <interface_name> command processor has timed-out waiting for a response from the NP interface task that initialization is complete.

Examples

```
start_np_interface in=cyber_109

NP interface CYBER_109 started.
```

START_OUTCALL_GATEWAY (STAOG) NETOU Command

Purpose Starts the Outcall Gateway.

Format START_OUTCALL_GATEWAY

Parameters None.

Responses Outcall Gateway is started.

--WARNING-- Outcall Gateway is already started.

--ERROR-- Outcall Gateway is not defined.

Examples

```
start_outcall_gateway

Outcall Gateway is started.
```

START_PASSTHROUGH_SERVICE (STAPS) NETOU Command

Purpose Starts the interactive passthrough service. The service allows passthrough ports to connect to the Interactive Passthrough Gateway and register their respective titles.

NOTE

In future releases, this command will no longer be supported. You should start using the START_OUTCALL_GATEWAY command as soon as possible.

Format **START_PASSTHROUGH_SERVICE**

Parameters None.

Responses Passthrough Service started.

--ERROR-- Passthrough Service not defined or already started.

Examples

```
start_passthrough_service

Passthrough Service started.
```

START_PORT_TEST (STAPT)
NETOU Command

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.

NOTE

Do not run diagnostics on a port configured with a trunk or line speed of 56 Kbs, that is attached to a CIM which is also supporting other activity. The port diagnostics fail.

Format

START_PORT_TEST
DEVICE_NAME = name
REPEAT_PASS = integer 0..65535
SUCCESS_STATE = keyword value
LOGGING = boolean
STOP_ON_ERROR = boolean
LOOP_MODE = keyword value
MODEM_CLASS = integer 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 value 0 specifies that the test is run indefinitely until you stop the test by a STOP_PORT_TEST command described in this manual. Default is 1.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error causes 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 is left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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.

Keyword	Description
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. See 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 RS-449 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

Keyword	Description

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 raises RTS and DTR and checks 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. See the CDCNET Hardware Installation and 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	RS-232	CCITT	RS-449
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 first executes 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 is only used when LOOP_MODE=MODEM is selected, and it is required if you choose MODEM loopback, and if the port has not been configured or has been configured as a line with autorecognition. The following modem class table provides information about modem classes and speeds. There is no default value for this parameter. See the CDCNET Hardware Installation and Troubleshooting Guide for more information on loopback testing for modems.

Modem Type	Operating Mode	Maximum Speed (bps)	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 The following response is received if the test starts successfully.

```
PORT test started, version <version_number>.
CIM Slot number = <cim slot number>.
LIM Slot number = <lim slot number>.
PORT number = <port number>.
```

You receive the following response if the test has already been started.

```
--WARNING-- Device <device_name> test already started.
```

The following error messages may be received.

```
--ERROR-- Device <device_name> not installed in system.
```

```
--ERROR-- Device <device_name> not in "DOWN" state.
```

```
--ERROR-- Device <device_name> cannot be tested at this time.
$CIM<cim slot number> must be dedicated to the <driver_type>
driver and is currently active. The STAPT command may be reissued
once all active lines/trunks on $CIM<cim slot number> are stopped.
```

```
--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*>.
```

You receive the following response if the test task could not start.

```
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) through 0F (16))
RS449 (xx=00 (16) through 07 (16))
V.35 (xx=20 (16) through 27 (16))

Test not allowed for LIM type xx.
```

You receive the following response when you try a port test on an RS-232 LIM with an invalid ID type. Only LIM testing is allowed. The port test is allowed on RS-232 LIMs with an ID type of 09 through 0E (16).

```
Port test is not allowed for LIM type xx
ENTER "start_lim_test dn=$limx"
```

You receive the following response when the test task starts but terminates 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 autorecognition 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-rec line.
```

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 cannot 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 you enter a START_LIM_TEST command,an abort response is 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"
```

Remarks In order for the port test to run, the device state must be DOWN. Use the CHANGE_ELEMENT_STATE (CHAES) command described in this manual to change the state of the device.

To get the results of the port test, send the DISPLAY_TEST_STATUS command described in this manual, 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

```
start_port_test device_name=$lim3_port1

PORT TEST STARTED, VERSION 10H3.
CIM slot number = 5.
LIM Slot number = 3.
Port number = 1.
```

START_PROCESS_METRICS (STAPM) NETOU Command

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 table lists the supported software processes that can be used as keywords. A note in parentheses is a brief description of the information tracked. The table also provides the log/alarm number associated with each process.

Software Process	Log/Alarm Message
command	405
directory	95
dod_internet	1357
independent_log_me (source log)	446
intranet (summary)	889
intranet (expanded)	890
log_support_application (source log)	424
osa (Operator Support Application)	547
session	737
system	299
tcp (Transmission Control Protocol)	1453
egp(tcp/ip external gateway protocol)	1435
clns	1648
es_is_routing	1873
channel_connection	1693
udp (tcp/ip user datagram protocol)	1700
transport (summary)	1628
transport (expanded)	1629
np_ivt_gw	746
is_is_routing	1913
cons	1924
llc_2	105

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)

Type of statistics group requested. Possible keyword values include the following:

SUMMARY (S)
EXPANDED (E)
DEBUG (D)
ALL (A)

Default is SUMMARY.

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

You receive the following response if the process metrics successfully started.

<process_name> <group_name> metrics 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. When you receive this message, you may enable any messages listed using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

Reporting log message <message_number> not enabled.

For processes that are not defined or do not support process metrics, the following lines are inserted:

<process> unknown.

<process _name> <group_name> metrics not supported.

Specified group not suppported for this process.

--FATAL-- <process _name> metrics failed.

--FATAL-- <process _name> metrics startup failed, not enough memory currently exists for required table space.

Remarks To check if these log messages are enabled, use the DISPLAY_SOURCE_LOG_GROUP command described in this manual. If these log messages are not enabled, enable them using the CHANGE_SOURCE_LOG_GROUP command described in this manual.

See the NPA commands in this manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
start_process_metrics p=session,ri=360,g=(summary,expanded)

SESSION summary metrics started
SESSION expanded metrics started
```

START_SERVER_TELNET_GW (STASTG) NETOU Command

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 described in this manual.

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

```
start_server_telnet_gw gateway_name=gw_to_cyber

Server TELNET gateway GW_TO_CYBER is started.
```

START_SLIP_GW (STASG)
NETOU Command

Purpose Starts the SLIP gateway service. The titles from the DEFINE_SLIP_GW command are registered so that CDCNET can establish SLIP networks to CDCNET.

Format

START_SLIP_GW
GATEWAY_NAME = name

Parameters

GATEWAY_NAME (GN)

The logical name of the SLIP gateway as previously defined in a DEFINE_SLIP_GW command. This parameter is required.

Responses

SLIP gateway <gateway_name> is started.

--ERROR-- SLIP gateway <gateway_name> is not defined.

--ERROR-- SLIP gateway title <title> is already defined.

--WARNING-- SLIP gateway <gateway_name> is already started.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
start_slip_gw gateway_name=slip_gw

SLIP gateway SLIP_GW is started.
```

START_SUBNET_CONNECT_TEST (STASCT) NETOU Command

Purpose Starts the inline diagnostic test between a directly connected source and destination system.

Format

```
START_SUBNET_CONNECT_TEST
   DESTINATION_SYSTEM = name
   SUBNET_NAME = name
   MESSAGE_LENGTH = keyword
   MESSAGE_COUNT = 1..60
   MESSAGE_INTERVAL = 0..1000
   REPEAT_PASS = integer 0..65535
   LOGGING = boolean
   STOP_ON_ERROR = boolean
```

Parameters

DESTINATION_SYSTEM (DS)

Specifies the system name to which the CLNS PDU is sent. The system name is designated by the DEFINE_SYSTEM command (described in this manual).

SUBNET_NAME (SN)

Specifies the subnet name used to reach the destination system. This name is designated by the DEFINE_xxx_NET commands (described in this manual). When the subnet name is not specified, the routing software determines the subnet used to reach the destination system.

MESSAGE_LENGTH (ML)

Specifies a random or fixed range of length for each message which is transmitted in the test. If a random value is selected, the length of the messages varies within the selected range as determined by the test. The value of the fixed range is from 1 to 512.

The following range value keywords maybe used:

Keyword	Range (bytes)
MAX	Maximum size per subnet
SMALL (S)	1 to 256
LARGE (L)	256 to MAX
MIXED (M)	1 to MAX

Default is MIXED.

MESSAGE_COUNT (MC)

Specifies the number of messages sent in each pass of the test. The values are from 1 to 60. Default is 1.

MESSAGE_INTERVAL (MI)

Specifies the time interval in milliseconds between transmissions of the messages. Zero indicates immediate transmission. The maximum time allowed between transmissions is 1 second (1000 milliseconds). Default is 0 millisecond.

This page was intentionally left blank.

REPEAT_PASS (RP)

Specifies the number of times the test is repeated. If 0 is used, the test runs continuously until stopped by the operator. Default is 1.

LOGGING (L)

Specifies whether log messages are issued (ON) or not (OFF). Default is ON.

STOP_ON_ERROR (SOE)

Specifies whether the test stops when an error is detected (ON) or if it continues running (OFF) until the pass counter is reached or an operator stops the test. Default is ON.

Remarks The source system is always the system to which the SENC command is issued and in which the STASCT is executing.

Responses Subnet Connect Test STARTED.
Destination System = <destination_system>
Subnet = <subnet_name>

--INFORMATIVE-- Subnet <subnet_name> is not available for testing.

--ERROR-- Subnet Connect Test of <subnet_name> is already running.

--ERROR-- Subnet <subnet_name> is congested, the test may not be started.

--ERROR-- System <destination_system> is unknown.

--ERROR-- <system_type> system <system_name> is NOT a CDCNET OSI system.

--ERROR-- Subnet <subnet_name> is not directly connected.

--ERROR-- System <destination_system> is not directly connected via subnet <subnet_name>. The Subnet Connect Test only allows directly connected systems to be tested.

--ERROR-- System <destination_system> is not directly connected. The Subnet Connect Test only allows directly connected systems to be tested.

--ERROR-- Unable to start test task.

--ERROR-- NIL pointer for status record in the NIB.

--ERROR-- No test task response before a timeout.

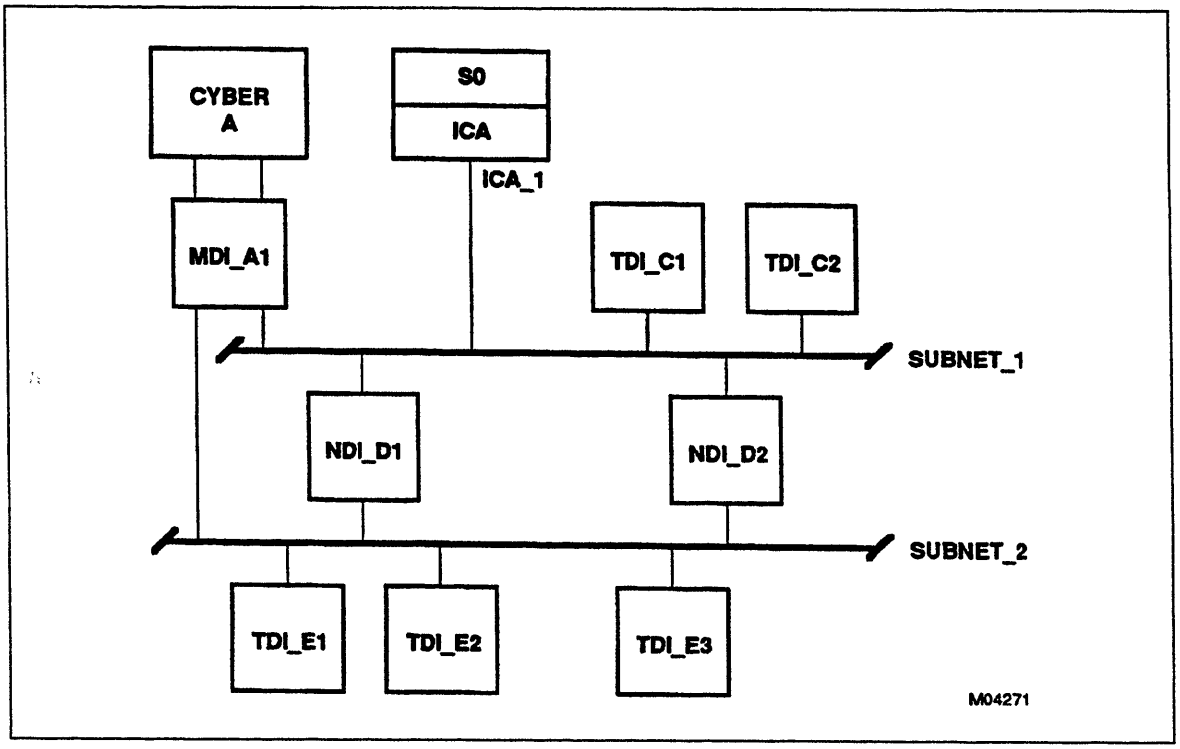


Figure 12-1. Catenet Example

Examples

Figure 12-1 shows a hypothetical catenet consisting of two subnets connected by two NDIs. The connection from CYBER A through MDI_A1 to both Subnet_1 and Subnet_2 illustrates dual paths from a system. In the following examples, references are made to this catenet.

```
senc c='start_subnet_connect_test destination_system=ndi_d2' s=mdi_a1
```

FROM MDI_A1

```
Subnet Connect Test STARTED.
Destination System = NDI_D2
Subnet = SUBNET_2
```

In this example, there are two possible paths to the destination system. Since no subnet was specified in the command, the response indicates the subnet used to reach system NDI_D2.

```
senc c='stasct ds=ndi_d1 sn=subnet_1 rp=0' s=mdi_a1
```

FROM MDI_A1

```
Subnet Connect Test STARTED.
Destination System = NDI_D1
Subnet = SUBNET_1
```

In this example, a subnet_name is specified, so the test must test between the two directly connected DIs on SUBNET_1.

START_TCPIP_GW (STATG)
NETOU Command

Purpose Starts the TCP/IP application interface gateway. The gateway registers titles specified from the DEFINE_TCPIP_GW command described in this manual 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 described in this manual.

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

```
start_tcpip_gw gateway_name=ftp_gw

TCP/IP gateway FTP_GW is started to support TCP protocol.
```

START_TRACE_PROCESS (STATP)

Purpose Starts the collection of trace information for the given process.

Format

```
START_TRACE_PROCESS
   PROCESS = name
   ARRAY_SIZE = integer 1 .. 65535
   DATA_SIZE = integer 1 .. 65535
   FILTER = list 1 .. 4 of any
```

Parameters

PROCESS (P)

Specifies the name of the process. The following processes and information can be traced:

Process	Description
CHANNEL_CONNECTION	Protocol Data Units
DFA (Dependent File Access)	Intertask messages
DIRECTORY	Protocol Data Units and database updates
DOD_INTERNET	Protocol Data Units
ES_IS_ROUTING	Protocol Data Units and miscellaneous events
HDLC	Protocol Data Units and intertask messages
LLC_2	Protocol Data Units
RIP (Routing Information Protocol)	Protocol Data Units and events
SYSTEM	Task snapshot at 100-millisecond intervals
TRANSPORT	Protocol Data Units
X25_PL (X.25 Packet Level)	Protocol Data Units

ARRAY_SIZE (AS)

Specifies the size of the array where the trace information is kept. For example, if ARRAY_SIZE is 10, only the last 10 trace information entries are kept before reusing the entries. The bigger the ARRAY_SIZE value, the more memory is used. Only in rare instances should this parameter be set larger than 100. If possible, depending on problem being analyzed, the operator should stop the traced process after the needed trace is captured to avoid memory problems. The default is 100.

DATA_SIZE (DS)

Specifies the amount of variable data to keep. An odd number is rounded to the next even number. The bigger the DATA_SIZE value, the more memory is used. Only in rare instances should this parameter be set larger than 32. If possible, depending on problem being analyzed, the operator should stop the traced process after the needed trace is captured to avoid memory problems. The default is 32.

FILTER (F)

Specifies a filter to be used in tracing. Only what is stated in the filter is traced. The value you specify is process-dependent:

Process	Allowed Filter Value
CHANNEL_ CONNECTION	SAP name, based on a CC SAP opened in the DI (to determine available names, use command DISPLAY_ CHANNEL_CONNECT_STATUS).
DFA	File name as a string (can include wild cards). The trace does not include ITMs (Intertask Messages) associated with other file names.
DIRECTORY	Title name (or subset of a title name). This is not a precise filter, since it involves matching an ASCII string within a Protocol Data Unit (PDU) for the PDU events. For the internal data structure events, a subset of the title matches only when the subset is at the beginning of the title.
DOD_INTERNET	IP address (list of 4 octets). The trace includes each IP datagram with a matching source or destination IP address.
ES_IS_ROUTING	Peer MAC address string (may include wild cards). The trace includes only data that is to be sent to the peer MAC address for transmitted data. It includes the miscellaneous items. **NOTE** The string filter has a maximum size of 32. If a value greater than 32 is given, the trace uses only the first 32 characters.
TRANSPORT	SAP name as a string (can include wild cards), based on a SAP opened in the DI (to determine available names, use command DISPLAY_OSI_TRANSPORT_ STATUS). Use the SAP name for any Transport user, for example, to trace RBF GW traffic for a potential printer problem. You can also supply a local reference ID. The trace includes only Transport Protocol Data Units (TPDUs) associated with the specified reference ID.

X25_PL	Either a trunk name, as defined by DEFINE_X25_TRUNK, DEFINE_HDLC_TRUNK, or DEFINE_LLC_2_TRUNK, or a peer DTE address string (can include wild cards). If you use a trunk name, the trace includes only data associated with the specified trunk name. If you use a peer DTE address string, the trace includes only packets associated with the logical channel numbers associated with the specified peer DTE address.
All others	The parameter is ignored.

NOTE

The default is no filtering. There is limited error checking on filters, both for syntax and semantics. The filters that use wild cards are case-sensitive; for example, DFA opens a Transport SAP with the name 'Dep. File Access', so a filter of '*ep.*' matches it, but a filter of 'DEP*' does not match.

Responses

TRACE started for process <name>.

--ERROR-- TRACE entry point not found for process <name>.

--ERROR-- TRACE already started for process <name>.

--FATAL-- Not enough memory currently exists for required table space.

Examples

```
start_trace_process process=transport filter='Dep*'

TRACE started for process TRANSPORT.
```

START_TRUNK (STAT)
NETOU Command

Purpose Starts communications over an HDLC or Ethernet trunk. This command is the same as START_NETWORK, except for HDLC rotary (multiple-trunk) networks. The START_NETWORK command starts all the HDLC trunks defined for the HDLC network. The START_TRUNK command starts only the given HDLC trunk. Use the START_NETWORK command except for rotary HDLC networks and when you want to start specific trunks.

For an X.25 trunk, either DEFINE_X25_INTERFACE START=YES or START_X25_INTERFACE starts the X.25 interface and the underlying media. For this reason, START_TRUNK does not work with a defined X.25 trunk. The error returned when you try to use START_TRUNK with an X.25 trunk is:

ERROR-- Unknown trunk type.

Format

START_TRUNK
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

Specifies the logical name of the trunk assigned by the DEFINE command that configured the trunk.

Responses

Trunk <name> started.

--ERROR-- Trunk <name> already started.

--WARNING-- The 3A Command Processor has timed out waiting for response from SSR.

Please check network status for completion of request.

--ERROR-- Trunk <name> is not defined.

--ERROR-- A network must be defined for trunk <name> before it can be started.

--ERROR-- Unknown trunk type.

--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-- Unable to start task <entry_point_name>.

--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 statistics SAP.

Unable to open memory management SAP.

Examples

```
start_trunk trunk_name = menlo_park_trunk_1

Trunk MENLO_PARK_TRUNK_1 started.
```

START_TRUNK_METRICS (STATM)
NETOU Command

Purpose

Starts the collection, and optional reporting of statistics for one or more trunks.

If statistics are already enabled for the specified trunk, the statistics are immediately reported and the report period restarted for the selected items. The trunk statistics include statistics from the Stream Service Routine supporting the trunk. These statistics are available from the NPA reports and the DISPLAY_TRUNK_METRICS command described in this manual.

Format

START_TRUNK_METRICS
TRUNK_NAME = list 1..15 of name
REPORT_INTERVAL = 1..86400
GROUP = list 1..2 of keyword value
REPORT = boolean

Parameters

TRUNK_NAME (TN)

The logical names of one or more trunks for which statistics are to be collected.

REPORT_INTERVAL (RI)

The statistic reporting interval in seconds. The maximum interval is 24 hours (86,400 seconds).

GROUP (G)

Specifies the level of statistics to be collected. The following keyword values are allowed:

SUMMARY (S)
EXPANDED (E)
ALL (A)

Default is SUMMARY.

REPORT (R)

Specifies whether the statistics should be reported via log messages. Default is YES.

Responses The following response occurs for each trunk metric successfully started.

Trunk <trunk_name> <group> metrics are started

The following response appears if the log message to report the metric was not enabled.

Reporting log message <message number> not enabled.

The following lines are output for trunks that are not defined and started or do not support the metric.

Trunk <trunk_name> <group> metrics are not supported.

The following error responses may occur.

--ERROR-- A network or interface is not defined and started for trunk <trunk_name>.

--FATAL-- Trunk <trunk_name> metrics startup failed.

--FATAL-- Trunk <trunk_name> metrics startup failed, not enough memory currently exists for required table space.

Remarks In order for trunk statistics to be reported, one of 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 either message is enabled, use the DISPLAY_SOURCE_LOG_GROUP command described in this manual. To enable either message, use the CHANGE_SOURCE_LOG_GROUP command described in this manual.

See the NPA commands in this manual for information on creating statistics reports using the REFORMAT_CDCNET_LOG_FILE (REFCLF) and CREATE_CDCNET_ANALYSIS_REPORT (CRECAR) commands.

Examples

```
start_trunk_metrics tn= bld_3_trunk,ri=360,g=(summary,expanded)

Trunk BLD_3_TRUNK summary metrics  started
Trunk BLD_3_TRUNK expanded metrics  started
```

START_UDP_INTERFACE (STAUI)
NETOU Command

Purpose Starts the User Datagram Protocol (UDP) interface layer. UDP must have been previously defined by the DEFINE_UDP_INTERFACE command described in this manual.

Format **START_UDP_INTERFACE**

Responses UDP Interface is started.

--ERROR-- UDP interface is not defined.

--ERROR-- UDP interface is already started.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
start_udp_interface

UDP interface is started.
```

START_URI_TEST (STAUT)
NETOU Command

Purpose Starts the online diagnostics test on an individual unit record interface (URI).

Format

START_URI_TEST
DEVICE_NAME = name
REPEAT_PASS = integer 0..65535
SUCCESS_STATE = keyword value
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 0 specifies that the test is run indefinitely until you stop the test by a STOP_URI_TEST command described in this manual. Default is 1.

NOTE

If the STOP_ON_ERROR parameter is set to OFF, an error causes 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 is left upon successful completion of the diagnostic test. Possible values are ON and DOWN. ON specifies that the device state is set to ON if the test completes without error, but remains set to the DOWN state if the test detects an error. DOWN specifies that the state remains 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.

Keyword	Description
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. See the CDCNET Hardware Installation and 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

The following response is returned if the test starts successfully.

URI test started, version <version number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>

The following response occurs if the test is already started.

--WARNING-- Device <device_name> test already started.

The following response indicates that the requested device cannot be tested because the CIM supporting the URI to be tested is already dedicated to the HDLC256 or HDLC3x64 driver and the URI to be tested is configured to use another driver.

--ERROR-- Device <device_name> cannot be tested at this time.
$CIM <cim slot number> must be dedicated to the <driver_type> and is currently active. The STAUT command may be reissued once all active lines/trunks on $CIM <cim slot number> are stopped.

The following response indicates the test never started.

--FATAL-- URI test aborted, version <version_number>
CIM slot number= <cim slot number>
URI slot number= <uri slot number>
Unable to start test task

--ERROR-- Device <device_name> not installed in system.

--ERROR-- Device <device_name> not in "DOWN" state.

--ERROR-- Device <device_name> addresses a port that cannot be serviced.
More than 48 ports are attached to CIM <cim_slot_number>.
Ports beyond the 48th port attached to a CIM are not serviced.

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 does not allow you to start a lower level test such as a URI test. When a CIM fails, you cannot 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

```
start_uri_test device_name = $uri5

URI test started, version 2301
CIM slot number= 6
URI slot number= 5
```

START_USER_TELNET_GW (STAUTG) NETOU Command

Purpose Starts the user TELNET interactive terminal gateway service. Gateway titles are selected with the TITLE (T) parameter of the DEFINE_USER_TELNET_GW command described in this manual. 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 described in this manual.

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

```
start_user_telnet_gw gateway_name=vax_gw

User TELNET gateway VAX_GW is started.
```

START_X25_ASYNCTIP (STAXA)
NETOU Command

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

```
start_x25_asynctip trunk_name = telenet_2

X.25 AsyncTip support started for specified trunks.
```

START_X25_GW (STAXG)
NETOU Command

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 described in this manual, and reactivates any title or titles previously inactivated (deregistered) when the gateway was stopped. Although the titles were inactivated, they remain 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 described in this manual.

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

```
start_x25_gw gn=telenet_gw

X.25 gateway TELENET_GW is started.
```

START_X25_INTERFACE (STAXI)
NETOU Command

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, and 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 described in this manual.

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.

Examples

```
start_x25_interface in=telenet_inf

X.25 interface TELENET_INF started on trunk TELENET2.
```

START_X25_TERMINAL_GW (STAXTG) NETOU Command

Purpose Starts the specified X.25 terminal gateway. If ADDXTGO commands have been entered for the associated gateway name or the ODR parameter is FALSE, it registers the title defined for the gateway in the CDCNET directory.

Format

```
START_X25_TERMINAL_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

Specifies the logical name of an X.25 terminal gateway defined by a DEFXTG command.

Responses

X.25 terminal gateway <name> is started.

X.25 terminal gateway <name> is started and awaiting outcall definitions.

--ERROR-- X.25 terminal gateway <name> is already started.

--ERROR-- X.25 terminal gateway <gateway_name> is not defined.

Examples

```
start_x25_terminal_gw gn=datapac_terminal_gateway

X.25 terminal gateway DATAPAC_TERMINAL_GATEWAY is started.
```

STOP Commands 13

STOP Commands 13

This chapter provides complete descriptions of all CDCNET Network Operations STOP commands. Command descriptions are organized alphabetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

STOP_CIM_TEST (STOCT)
NETOU Command

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. Allowed range is $CIM 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, described in this manual, to the DI that contains the device being tested.

Examples

```
stop_cim_test device_name=$cim5

CIM test flag set, version 7606
CIM slot number = 5.
```

STOP_EGP_PEER (STOEP)
NETOU Command

Purpose Stops the exchange of routing information with the specified peer. Begins Exterior Gateway Protocol (EGP) 'cease' processing with the peer, if the peer is acquired. If the peer is not acquired and is defined as a backup peer, then the specified peer is removed from the eligibility as a backup. This command reverses the effect of the START_EGP_PEER command described in this manual.

Format

STOP_EGP_PEER
IP_ADDRESS = list 4 of 0..255

Parameters

IP_ADDRESS (IA)

Identifies the IP address of the EGP peer gateway with which exchange of routing information is to be stopped.

Responses

EGP peer <ip_address> is stopped.

EGP peer <ip_address> is not idle. 'Cease' process initiated.

--WARNING-- EGP peer <ip_address> is already stopped.

--ERROR-- EGP peer <ip_address> is not defined.

Examples

```
stop_egp_peer ip_address = (128,5,0,3)

EGP peer 128.5.0.3 is stopped.
```

STOP_ESCI_TEST (STOET)
NETOU Command

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 described in this manual to the DI that contains the device being tested.

Examples

```
stop_esci_test device_name=$esci6

ESCI test flag set, version 7606
ESCI slot number = 6.
```

STOP_LIM_TEST (STOLT)
NETOU Command

Purpose Stops the online diagnostics running on a LIM board and its ports.

Format

```
STOP_LIM_TEST
   DEVICE_NAME = name
```

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).

Responses You receive one of the following responses when this command is executed.

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.
CIM Slot number = <cim_slot_number>.
LIM Slot number = <lim_slot_number>.

Remarks To get the results of the LIM test, send the DISPLAY_TEST_STATUS command described in this manual to the DI that contains the device being tested.

Examples

```
stop_lim_test device_name=$lim2

LIM test flag set, version 7606
CIM Slot number = 5.
LIM Slot number = 2.
```

STOP_LINE (STOL)
NETOU Command

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

```
stop_line line_name=engin_bld_31

Line ENGIN_BLDG_31 stopped.
```

STOP_LINE_METRICS (STOLM)
NETOU Command

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

LINE_NAME (LN)

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 the following:

SUMMARY (S)
EXPANDED (E)
ALL (A)

Default is ALL.

Responses You receive the following response if the line metrics successfully stopped.

Line <line_name> <group_name> metrics stopped.
(One response for each line and group specified in the command.)

--WARNING-- Future CDCNET releases will only support start/stop trunk metrics commands. Please replace start/stop line metrics commands with start/stop trunk metrics commands.

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

```
stop_line_metrics line_name=(line_303,line_305,..
line_306,line_310),g=(summary,expanded)

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) NETOU Command

Purpose Stops the inline diagnostics test executing on an MCI board.

Format

```
STOP_MCI_INLINE_TEST
   DEVICE_NAME = name
```

Parameters

DEVICE_NAME (DN)

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

```
stop_mci_inline_test dn=$mci6

Stopped the MCI in line test
for device $mci6
```

STOP_MCI_TEST (STOMT)
NETOU Command

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

```
stop_mci_test dn=$mci6

MCI test stop flag set, version 7606
MCI slot number= 6
```

STOP_NET_DELAY_MEASUREMENT (STONDM) NETOU Command

Purpose Stops execution of the network delay measurement. The internal control flag, STOP_FLAG, is set causing the measurement to be terminated when the task next checks the flag. The flag is tested by the measurement task once a second to allow the user to stop a task without waiting for any lengthy timeouts.

Format

STOP_NET_DELAY_MEASUREMENT
DESTINATION_SYSTEM = name

Parameters

DESTINATION_SYSTEM (DS)

Specifies the system name, which the network delay measurement is to be made, from the source (originating) system. This is the name as designated by the DESTINATION_SYSTEM parameter on the START_NET_DELAY_MEASUREMENT command.

Responses

Network Delay Measurement to <destination_system> is being stopped.

--ERROR-- Network Delay Measurement to <destination_system> is not active.

--ERROR-- System <destination_system> is unknown.

--ERROR-- Destination system <destination_system> is not a CDCNET OSI system.

Examples

```
senc 'stop_net_delay_measurement destination_system=ndi_1' s=mdi_a1

FROM MDI_A1

Network Delay Measurement to NDI_1 is being stopped.
```

STOP_NETWORK (STON)
NETOU Command

Purpose Stops communications over a network solution, such as Ethernet, X.25, or HDLC. For Ethernet or HDLC network, STOP_NETWORK also stops the underlying 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 described in this manual.

A STOP_NETWORK command is the same as a STOP_TRUNK command, except for HDLC rotary trunks. Both STOP_NETWORK and STOP_TRUNK stop the communication media. A STOP_TRUNK command does not work for X.25 trunks; only the STOP_NETWORK or STOP_X25_INTERFACE command stops the media of communication for X.25. The only time it makes sense to use STOP_TRUNK is for HDLC rotaries. To stop only one of the HDLC rotary trunks, use STOP_TRUNK. If all HDLC rotary trunks are to be stopped, use STOP_NETWORK .

Format

STOP_NETWORK
NETWORK_NAME = name

Parameters

NETWORK_NAME (NN)

The logical name of the network assigned by a define command.

Responses

<Network_type> network <name> stopped for trunk <trunk_name>.

<Network_type> network <name> stopped for trunks (<trunk_name> <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

```
stop_network network_name=tymnet_net_1

X.25 Network TYMNET_NET_1 stopped for trunk TYMNET_TRUNK1.
```

STOP_NETWORK_METRICS (STONM)
NETOU Command

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 (S)
EXPANDED (E)
ALL (A)

Default is ALL.

Responses You receive the following response if the network metrics successfully stopped.

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

```
stop-network_metrics ..
network_name=bld_3_ethernet,group=(summary,expanded)

Network BLD_3_ETHERNET summary metrics stopped.
Network BLD_3_ETHERNET expanded metrics stopped.
```

STOP_NP_GW (STONG) (NOS MDI Only) NETOU Command

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 described in this manual. 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 described in this manual.

Responses NP gateway <gateway_name> stopped.

--ERROR-- NP gateway <name> not defined or already stopped.

Examples

```
stop_np_gw gn=a_to_a_109 s=mdi2

NP gateway A_TO_A_109 stopped.
```

STOP_NP_INTERFACE (STONI) (NOS MDI Only) NETOU Command

Purpose Stops the Network Products (NP) 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

```
stop_np_interface in=cyber_109

NP interface CYBER_109 stopped.
```

STOP_NP_TERMINAL_GW (STONTG) NETOU Command

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 described in this manual that configured the gateway.

Responses NP terminal gateway <name> stopped.

--ERROR-- NP terminal gateway name not defined or already stopped.

Examples

```
stop_np_terminal_gw gn=ivt109

NP terminal gateway IVT109 stopped.
```

STOP_OUTCALL_GATEWAY (STOOG)
NETOU Command

Purpose Stops the Outcall Gateway. Stopping the Outcall Gateway causes all existing DEVICE_OUTCALL and Passthrough connections in the DI to be terminated.

Format STOP_OUTCALL_GATEWAY

Parameters None.

Responses Outcall Gateway is stopped.

--WARNING-- Outcall Gateway is already stopped.

--ERROR-- Outcall Gateway is not defined.

Examples

```
stop_outcall_gateway

Outcall Gateway is stopped.
```

STOP_PASSTHROUGH_SERVICE (STOPS) NETOU Command

Purpose Stops the Interactive Passthrough Service. This command terminates all existing passthrough connections in the DI.

NOTE

In future releases, this command will no longer be supported. You should start using the STOP_OUTCALL_GATEWAY command as soon as possible.

Format **STOP_PASSTHROUGH_SERVICE**

Parameters None.

Responses Passthrough Service stopped.

--ERROR-- Passthrough Service not defined or already stopped.

Examples

```
stop_passthrough_service

Passthrough Service stopped.
```

STOP_PORT_TEST (STOPT)
NETOU Command

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 (described in this manual) to the DI that contains the device being tested.

Examples

```
stop_port_test device_name=$lim3_port2

Port test stop flag set, version 7606
CIM slot number = 5
LIM slot number = 3
Port slot number = 2
```

STOP_PROCESS_METRICS (STOPM) NETOU Command

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 remain in effect.

Format

STOP_PROCESS_METRICS
PROCESS = list 1..15 of name
GROUP = list 1..3 of keyword value

Parameters

PROCESS (P)

Specifies the name of the communications system process for which statistics collection and reporting is to stop. The following keywords are supported.

command
directory
dod_internet
independent_log_me (source log)
intranet (summary)
intranet (expanded)
log_support_application (source log)
osa (Operator Support Application)
session
system
tcp (Transmission Control Protocol)
tcp/ip external gateway protocol (EGP)
clns
es_is_routing
channel_connection
tcp/ip user datagram protocol (udp)
transport (summary)
transport (expanded)
np_ivt_gw
is_is_routing
cons
llc_2

GROUP (G)

Specifies the level of statistics collection to stop. Specify one of the following levels:

SUMMARY (S)
EXPANDED (E)
DEBUG (D)
ALL (A)

Default is ALL.

Responses You receive the following response if the process metrics successfully stopped.

<process> <group> metrics stopped

For process metrics not stopped because of errors during command entry, processes that are unknown, or processes for which the metric was not started, the following lines are output.

Process <process> unknown

<process> <group> metrics not started

--FATAL-- <process> metrics shutdown failed

Examples

```
stop_process_metrics p=intranet,g=(summary,expanded)

INTRANET summary metrics stopped
INTRANET expanded metrics stopped
```

STOP_SERVER_TELNET_GW (STOSTG) NETOU Command

Purpose Stops the host terminal gateway service. The gateway terminates any established connections and stops listening for new connections. This commandmanual reverses the effect of a START_SERVER_TELNET_GW command described in this manual.

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 described in this manual.

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

```
stop_server_telnet_gw gateway_name=gw_to_cyber

Server TELNET gateway GW_TO_CYBER is stopped.
```

STOP_SLIP_GW (STOSG)
NETOU Command

Purpose Stops the SLIP gateway service. The gateway terminates any established connections and takes titles associated with this gateway interface out of registration. This command reverses the effect of a START_SLIP_GW command.

Format

```
STOP_SLIP_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

The logical name of the SLIP gateway previously defined in a DEFINE_SLIP_GW command. This parameter is required.

Responses

SLIP gateway <gateway_name> is stopped.

--WARNING-- SLIP gateway <gateway_name> is already stopped.

--ERROR-- SLIP gateway <gateway_name> is not defined.

Examples

```
stop_slip_gw gateway_name=slip_gw

SLIP gateway SLIP_GW is stopped.
```

STOP_SUBNET_CONNECT_TEST (STOSCT) NETOU Command

Purpose Stops the test execution of the online diagnostic started by the STASCT command described earlier in this manual.

Format

STOP_SUBNET_CONNECT_TEST
SUBNET_NAME = name

Parameters

SUBNET_NAME (SN)

Specifies the name of the subnet being tested by the STASCT command.

Responses

Subnet Connect Test on <subnet_name> is being stopped.

--ERROR-- Subnet <subnet_name> is unknown.

--ERROR-- Subnet Connect Test on <subnet_name> is not running.

Examples

```
stop_subnet_connect_test sn=SUBNET_1

Subnet Test on SUBNET_1 is being stopped.
```

STOP_TCPIP_GW (STOTG)
NETOU Command

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 described in this manual.

Responses

TCP/IP gateway <gateway_name> is stopped.

--WARNING-- TCP/IP gateway <gateway_name> is already stopped.

--ERROR-- TCP/IP gateway <gateway_name> is not defined.

--FATAL-- TCP/IP gateway was unable to deregister title <title>.

Examples

```
stop_tcpip_gw gateway_name=ftp_gw

TCP/IP gateway FTP_GW is stopped.
```

STOP_TRACE_PROCESS (STOTP)

Purpose Stops the collection of trace information for the given process.

Format

```
STOP_TRACE_PROCESS
   PROCESS = name
```

Parameters

PROCESS (P)

Specifies the name of the trace process to be stopped.

Responses

TRACE stopped for process <name>.

--ERROR-- TRACE not active for process <name>.

Examples

```
stop_trace_process process=transport

TRACE stopped for process TRANSPORT.
```

STOP_TRUNK (STOT)
NETOU Command

Purpose Stops communications over an HDLC, X25, or Ethernet trunk. This is the same as STOP_NETWORK except for HDLC rotary (multiple trunk) networks. The STOP_NETWORK command stops all the HDLC trunks defined for the HDLC network. The STOP_TRUNK command stops only the given HDLC trunk. Use the STOP_NETWORK command except for rotary HDLC networks and when you want to stop specific trunks.

For an X.25 trunk, the command STOP_X25_INTERFACE stops both the X.25 interface and the underlying media. For this reason, STOP_TRUNK does not work with a defined X.25 trunk. The error returned when you try to use STOP_TRUNK with an X.25 trunk is:

--ERROR-- Unknown trunk type.

Format

STOP_TRUNK
TRUNK_NAME = name

Parameters

TRUNK_NAME (TN)

Specifies the logical name of the trunk assigned by the DEFINE command that configured the trunk.

Responses

Trunk <name> stopped.

--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-- Trunk <name> not defined.

--ERROR-- Unknown trunk type.

--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.

HDLC SSR received error when sending command to DVM.

Examples

```
stop_trunk trunk_name = menlo_park_trunk_1

Trunk MENLO_PARK_TRUNK_1 stopped.
```

STOP_TRUNK_METRICS (STOTM)
NETOU Command

Purpose Stops the collection and reporting of statistics at a statistics level for one or more trunks. Statistics are immediately reported for the stopped trunk statistics. Any statistics not specifically stopped continue to be collected and reported.

Format

STOP_TRUNK_METRICS
TRUNK_NAME = list 1..15 of name
GROUP = list 1..2 of keyword value

Parameters

TRUNK_NAME (TN)

The logical names of one or more trunks 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 the following:

SUMMARY (S)
EXPANDED (E)
ALL (A)

Default is SUMMARY.

Responses The following response is sent for each trunk and group for which metrics were stopped.

Trunk <trunk_name> <group> metrics are stopped.

The following display contains the following lines for trunks that are not defined and started or the metrics that were never started.

Trunk <trunk_name> is not defined and started.

<trunk_name> <group> metrics are not started.

--ERROR-- A network or interface is not defined and started for trunk <trunk_name>.

--FATAL-- Trunk <trunk_name> metrics shutdown failed.

Examples

```
stop_trunk_metrics tn= bld_3_trunk,..
g=(summary,expanded)

Trunk BLD_3_TRUNK summary metrics are stopped.
Trunk BLD_3_TRUNK expanded metrics are stopped.
```

STOP_UDP_INTERFACE (STOUI)
NETOU Command

Purpose Stops the User Datagram Protocol (UDP) layer interface. Closes all UDP service access points (SAPs), after which no new open SAP requests are honored.

Format **STOP_UDP_INTERFACE**

Parameters None.

Responses UDP interface is stopped.

--WARNING-- UDP interface is already stopped.

--ERROR-- UDP interface is not defined.

Examples

```
stop_udp_interface

UDP interface is stopped.
```

STOP_URI_TEST (STOUT)
NETOU Command

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

```
stop_uri_test dn=$uri5

URI test stop flag set, version 2301
CIM slot number= 3
URI slot number= 5
```

STOP_USER_TELNET_GW (STOUTG)
NETOU Command

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 described in this manual.

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 described in this manual.

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

```
stop_user_telnet_gw gateway_name=vax_gw

User TELNET gateway VAX_GW is stopped.
```

STOP_X25_ASYNCTIP (STOXA)
NETOU Command

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 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

```
stop_x25_asynctip trunk_name = telenet_2

X.25 AsyncTip support stopped for specified trunks.
```

STOP_X25_GW (STOXG)
NETOU Command

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 described in this manual.

Responses

X.25 gateway <name> stopped.

--WARNING-- X.25 gateway <name> is already stopped.

--ERROR-- X.25 gateway <name> is not defined.

Examples

```
stop_x25_gw gateway_name=telenet_gw

X.25 gateway TELENET_GW stopped.
```

STOP_X25_INTERFACE (STOXI)
NETOU Command

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, and 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 described in this manual.

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

```
stop_x25_interface interface_name=telenet_if

X.25 interface TELENET_IF stopped on trunk TELENET2.
```

STOP_X25_TERMINAL_GW (STOXTG)
NETOU Command

Purpose Stops the X.25 terminal gateway. Also, deletes the title registered for the gateway in the CDCNET directory.

Format

```
STOP_X25_TERMINAL_GW
   GATEWAY_NAME = name
```

Parameters

GATEWAY_NAME (GN)

Specifies the logical name assigned to the gateway by a DEFXTG command.

Responses

X.25 terminal gateway <name> is stopped.

--WARNING-- X.25 terminal gateway <name> is already stopped.

--ERROR-- X.25 terminal gateway <name> is not defined.

Examples

```
stop_x25_terminal_gw gateway_name = datapac_terminal_gateway

X.25 terminal gateway DATAPAC_TERMINAL_GATEWAY is stopped.
```

STOP_X25_VIRTUAL_CIRCUIT (STOXVC)

NETOU Command

Purpose Stops the identified virtual circuit on the specified X.25 trunk to clear the circuit.

Format

STOP_X25_VIRTUAL_CIRCUIT
VIRTUAL_CIRCUIT = string or **integer**
TRUNK_NAME = name

Parameters

VIRTUAL_CIRCUIT (VC)

Specifies the virtual circuit number assigned to the X.25 virutal circuit that is to be stopped. This parameter can be specified as either a string or an integer. This parameter can be one of the following:

$CONSOLE name, which includes the *TRUNK_NAME* (and DI), specified as a string. If the *TRUNK_NAME* parameter is specified, it is validated against the *TRUNK_NAME* part of the console name.

The 10-digit number (base 10) as it appears in the DISPLAY_CONNECTION output display, specified as a string. This 10-digit number includes the *TRUNK_NAME* and the *VIRTUAL_CIRCUIT* number. If the *TRUNK_NAME* parameter is specified, it is validated against the *TRUNK_NAME* part of the 10-digit number.

Logical channel number (base 8, 10, or 16; the default is 10). An error is returned if the *TRUNK_NAME* parameter is not also specified.

TRUNK_NAME (TN)

Specifies the logical name of the X.25 trunk on which the virtual circuit to be stopped resides. If *TRUNK_NAME* is not specified, the trunk must be included in the *VIRTUAL_CIRCUIT* parameter.

SYNC, UNLF, UNLM, and WRITM Commands 14

SYNC, UNLF, UNLM, and WRITM Commands 14

The command descriptions in this chapter are organized alphabetically by full command name. Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

SYNCHRONIZE_CLOCK (SYNC) NETOU Command

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 (described in this manual) 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

Parameters None.

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

```
synchronize_clock

System clock synchronized.
```

UNLOAD_FILE (UNLF)
NETOU Command

Purpose Removes a CDCNET file from DI memory residency. The buffers occupied by the file are released to the system.

Format

UNLOAD_FILE
FILE = name or string 1..63
FILE_TYPE = name

Parameters

FILE (F)

The name of the file to be unloaded. File names entered as strings are converted to uppercase and are restricted to the character set of SCL type name.

FILE_TYPE (FN)

Specifies the type of file to be loaded. Valid file types include the following:

EXCEPTION_LIST
TERMINAL_DEFINITION_PROCEDURE
TERMINAL_USER_PROCEDURE
DEVICE_LOAD_PROCEDURE
VALIDATION

Default is USER_PROCEDURE.

Responses File <name> is unloaded.

--WARNING-- File <name> is not currently loaded.

Remarks The format of VALIDATION file names is 'username@domain'. Username and domain are 1- through 31-character names. Both username and domain must be specified.

Examples

```
unload_file f=printer_with_vfu_p31310

File printer_with_vfu_p31310 is unloaded.
```

UNLOAD_MODULE (UNLM)
NETOU Command

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
ENTRY_POINT = name

Parameters *MODULE (M)*

The name of the desired software module.

ENTRY_POINT (EP)

The entry point name used to determine which module to make a candidate for unloading.

Responses Module <module> retain removed.

--ERROR-- Entry point <entry point> is not contained within module <module>.

--ERROR-- Module for entry point <entry point> was not found.

--ERROR-- Module <module> not currently loaded.

--ERROR-- Module <module> not previously retained.

--ERROR-- One of the parameters Module or Entry Point must be specified.

Remarks To display the modules currently loaded and/or marked for unloading in a DI, send the DISPLAY_SOFTWARE_LOAD_STATUS command described in this manual to the DI.

Examples

```
unload_module m=display_hardware_status

Module CMD_DISPLAY_HARDWARE_STATUS retain removed.
```

WRITE_TERMINAL_MESSAGE (WRITM) NETOU Command

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 to only 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 is sent as a string value within SEND_COMMAND, you must begin and end the message with two consecutive apostrophes so that the message is 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. It also specifies the name of a TELNET Server.

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 output from a working connection is ready for a terminal, the message is inserted in the output. If the terminal has multiple working connections, the message appears 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 described in this manual.

Examples

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

Message written.
```

Terminal Definition/User Procedure Commands 15

Terminal Definition/User Procedure Commands 15

This chapter contains descriptions of commands used in terminal definition procedures (TDPs) and terminal user procedures (TUPs).

These commands can be executed only by executing a TDP or TUP containing the commands. These commands cannot be entered individually through the Network Operator Utility (NETOU) or by the terminal user.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

See the CDCNET Terminal Interface manual for the complete set of TUP commands.

DEFINE_ACCESSIBLE_REMOTE_SYSTEM (DEFARS) Terminal Definition Procedure Command

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 described in this manual. This command may be executed only by inclusion in a TDP.

Format

DEFINE_ACCESSIBLE_REMOTE_SYSTEM
ACCESSIBLE_REMOTE_SYSTEM_NAME = name
LINE_NAME = name
ACCESSIBLE_REMOTE_SYSTEM_TYPE = keyword value
ROUTE_BACK_POSITION = 0..254 or *keyword value*
AUTHORITY_LEVEL = keyword value

Parameters

ACCESSIBLE_REMOTE_SYSTEM_NAME (ARSN)

Specifies the logical name of a remote system that 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, (described in this manual) the DEFARS command is ignored. Default is the name of the activating line.

ACCESSIBLE_REMOTE_SYSTEM_TYPE (ARST)

Specifies the type of accessible remote system (mainframe and/or operating system). The following keyword values are allowed for this parameter:

NOS_VE
NOS
NOS_BE
IBM
DEC
CYBER_205
ETA
CRAY
USER

Default is IBM.

This parameter is ignored for NJE remote systems. NTF uses this information to determine the routeback information format to insert when sending jobs to the accessible remote system. Routeback is the mechanism by which NTF propagates file ownership, destination, and routing characteristics.

ROUTE_BACK_POSITION (RBP)

Specifies the position in the job after which routeback information is inserted by NTF. Specifying NONE indicates no routeback information is inserted. Specifying 0 indicates routeback information is inserted as the first record of the job. This parameter is ignored for NJE remote systems. Defaults depend on the value of ACCESSIBLE_REMOTE_SYSTEM_TYPE, and are as follows:

ACCESSIBLE_REMOTE_SYSTEM_TYPE	Default Routeback Position
NOS_VE	3
NOS	3
NOS_BE	3
IBM	2
DEC	2
CYBER_205	3
ETA	2
CRAY	3
USER	NONE

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 <name> 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)
Terminal Definition Procedure Command

Purpose

Defines a batch device on a configured I/O station. Can be executed only by inclusion in a TDP. Some parameters do not apply to batch devices for exclusive NOS use, so NOS/VE and NOS formats are listed separately.

Configuring a device as a batch device implies that the device's I/O is supported through Batch Transfer Protocol (BTP). This command allows a device to be connected through 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.

For printers supported by NOS Printer Support Utility (PSU), some parameters are controlled by PSU rather than DEFINE_BATCH_DEVICE (see Remarks section).

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..176
PAGE_WIDTH = 0..255
TERMINAL_MODEL = name
TRANSMISSION_BLOCK_SIZE = 128..4095
CODE_SET = name or *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
CONTROL_CODE_REPLACEMENT = list 1..64, 1..2 range of <ccode>
DATA_PARITY = keyword value

START = boolean
TRAILER_PAGE = boolean

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 = 0..255
TERMINAL_MODEL = name
TRANSMISSION_BLOCK_SIZE = 128..4095
CODE_SET = name or *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
CONTROL_CODE_REPLACEMENT = list 1..64, 1..2 range of <ccode>
DATA_PARITY = keyword value
START = boolean
TRAILER_PAGE = boolean

Parameters **DEVICE_NAME (DN)**

Specifies the logical name of the batch device.

For batch devices that connect to NOS Remote Batch Facility (RBF), 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 used by RBF to form the RBF logical device name (that is, LPn, CRn, PUn).

If the DEVICE_NAME parameter is longer than seven characters, or contains an underscore, the RBF terminal name is generated by using the first seven characters, or all the characters before the first underscore if it is one of the first seven characters. For example, LINE1_PRINTER has a NOS RBF terminal name of LINE1 (RBF logical device name LP1, assuming device_type=printer), and READER2_LINE06 has a NOS RBF terminal name of READER2 (RBF logical device name CR2, assuming device_type=reader).

If you need to identify which line is associated with an RBF terminal name (for example, the NOS UCLP accounting message), a unique batch DEVICE_NAME must be specified for each batch device for each line.

For batch devices supported by Printer Support Utility (PSU), the same restrictions apply, except that the digit is not used for a device ordinal.

LINE_NAME (LN)

Specifies the logical name of the line to which the device is connected. If a DEFBD command is part of a TDP executed by line activation (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)

This parameter is used for BSC3270 and MODE4 TIPs. Specifies the cluster address to be used by the TIP for communication with the batch device. Default is 0.

For the HASP TIP, only one cluster is allowed on each line so this parameter is not used.

For the MODE4 TIP, the CLUSTER_ADDRESS must be in the range of 70(16) through 7F(16) for Mode 4A and 20(16) through 7F(16) for Mode 4C clusters. Since the default is 0, this parameter must be specified for all MODE4 devices.

For the BSC3270 TIP, the cluster address must be in the range of 0 through 31. Default is 0.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the cluster address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

DEVICE_ADDRESS (DA)

This parameter is used for HASP, BSC3270 and MODE4 TIPs. Specifies the device address to be used by the TIP for communication with the batch device. Default is 0.

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.

For the MODE4 TIP, the DEVICE_ADDRESS must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. Since the default is 0, this parameter must be specified for all MODE 4 devices.

For the BSC3270 TIP, the device address must be in the range of 0 through 31. Default is 0.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the device address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

DEVICE_TYPE (DT)

Specifies the type of batch device. The following device types are supported.

READER (R)
PRINTER (PR)
PUNCH (PU)
PLOTTER (PL)

Default is PRINTER.

The TIP must be able to support the specified device type; otherwise this command is rejected. For example, an error is 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
NJEF	None
URI	PRINTER
XPC	None
NTF	None
SNA3270	CONSOLE
X.25 ASYNCTIP	PRINTER

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)

Default 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, (described in this manual) 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 is sent to a printer following an output file unless the TRAILER_PAGE parameter is set to TRUE. Default is 1.

In a future CDCNET release, this interdependence of BPC and TP will be removed; BPC will have no effect on whether or not accounting information is printed.

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)

Default is POST_PRINT.

This parameter is ignored by the Asynchronous and URI TIPs. For devices supported by the BSC3270 TIP, only PRE_PRINT is supported.

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 (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, default for this parameter is NORMAL. For any other device type, 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.)

For devices supported by the BSC3270 TIP, parameters EC1 through EC4 do not apply.

EXTERNAL_CHARACTERISTICS_2 (EC2)

Specifies the second external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

EXTERNAL_CHARACTERISTICS_3 (EC3)

Specifies the third external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

EXTERNAL_CHARACTERISTICS_4 (EC4)

Specifies the fourth external device characteristic string supported by this device. See EXTERNAL_CHARACTERISTICS_1 for more information.

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 only for PRINTER device types. Forms codes are used to select the files that may be printed on the device. Default 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 only for PRINTER device types. 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 only for PRINTER device types. 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 only for PRINTER device types. 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 only for PRINTER, PLOTTER, or PUNCH device types. 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. This parameter has been replaced by the FORMS_SIZE parameter. For compatibility, this parameter is allowed on the DEFINE_BATCH_DEVICE command, but is 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. If a value of 0 is specified, and a batch transparent output file is sent to a HASP device, the PW value is interpreted as infinite.

Default for PRINTER devices is 136 columns. Default for PUNCH or PLOTTER device types is 80 columns.

TERMINAL_MODEL (TM)

Specifies a 1- to 31-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 Terminal Model
ASYNC	CDC_537V
HASP	CDC_CYBER18
URI	CDC_585V
MODE4	M4IMP
X.25 ASYNCTIP	CDC_537V

Other Control Data-supplied values for printers that you may specify for this parameter include the following:

Value	Description
CDC_533V_536V	Control Data 533 or 536 printer with VFU load image.
ASYNC_ PRINTER_ WITHOUT_VFU	Asynchronous printer without VFU load image.
M4NIMP	Mode 4 nonimpact printer.
POSTSCRIPT	PostScript printer.
XEROX_SPUR	Xerox laser 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 manual.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size used by the TIP for communication with the batch device. Use this parameter for output devices (such as consoles or printers).

The value on this command overrides the TRANSMISSION_BLOCK_SIZE parameter on the DEFINE_LINE command (described in this manual) for this device only. Default is the value specified on the DEFINE_LINE command described in this manual. This parameter is ignored by the URI TIP.

CODE_SET (CS)

Identifies the batch device's code set. You may specify a CDCNET-supplied code set or a site-defined code set. To specify a CDCNET-supplied code set, use one of the following keyword values:

ASCII
ASCII48
ASCII64
ASCII95
ASCII128
ASCII256
EBCDIC

Site-defined code sets have names selected by a site.

When a name rather than a keyword value is specified, it is assumed that a site-defined translation table is used. An attempt is made to locate a load procedure with a procedure name matching the code set name. For input devices, the code set name, as specified by the load procedure name, has the suffix @INPUT. Do not include this suffix should as part of the code set value for this parameter.

When one of the keyword values is specified, a Control Data-defined internal translation table is used. These tables are described in detail in the CDCNET Configuration Guide.

The default, and supported, code sets depend upon the TIP supporting the batch device:

TIP	Supported CDCNET-Supplied Code Sets
ASYNC and X.25 Asynchronous	ASCII128 (default), ASCII256 (if DATA_PARITY=NONE) ASCII, ASCII48, ASCII64, ASCII95
BSC3270	EBCDIC (default)
HASP	EBCDIC (default)
URI	ASCII95 (default), ASCII, ASCII48, ASCII64, ASCII128, ASCII256
MODE4	ASCII95 (default), ASCII64

ASYNC, X.25 Asynchronous, BSC3270, HASP, and URI support site-defined translation tables for batch output. HASP supports a site-defined translation table for batch input.

VFU_LOAD_PROCEDURE (VLP)

Specifies the name of the procedure containing the default VFU load image for the batch device. If the value for the VFU_LOAD_OPTION is any value but NONE, default for VLP is the Control Data-supplied procedure CDC_VFU for user-defined terminal models. For other Control Data-defined models, the default is VFU_*model_name*.

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 specifying 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 or not the density for the device can be changed by the TIP. The following values are allowed:

Keyword Value	Description
SIX_ANY, EIGHT_ANY	The vertical print density can be changed to either six lines per inch or eight lines per inch.
SIX_ONLY, EIGHT_ONLY	The vertical print density, cannot be changed.

Default for HASP and MODE4 TIPs is SIX_ONLY. Default for the URI, ASYNC, and X.25 Asynchronous TIPs is SIX_ANY. For terminal model POSTSCRIPT, VPD is forced to SIX_ONLY. For terminal models starting with 'POSTSCRIPT_', SIX_ONLY is the default.

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 0.5 to 31 inches are allowed. This parameter replaces the PAGE_LENGTH parameter. The forms size value is passed to the control facility, and is used, with the file-specified vertical print density, to select files for printing. Default is 11.

FORMS_SIZE is automatically adjusted to account for margins used on page printers for batch devices specifying one of the following values for the TERMINAL_MODEL parameter:

POSTSCRIPT
POSTSCRIPT_ (site defined name beginning with POSTSCRIPT_)
XEROX_SPUR

Specifically, the following adjustments are made for FORMS_SIZE parameter values specified on the DEFBD command:

Specified Value	Actual Value
Less than 1.5	0.5
Less than 12.0 and greater than 1.5	FORMS_SIZE - 1
More than 12.0	FORMS_SIZE - 2

NOTE

When printing files in horizontal orientation with the XEROX_SPUR printer model, if the FORMS_SIZE is greater than 11.0, the actual number of lines that print on a page may not be equal to FORMS_SIZE times vertical print density.

If you specify a FORMS_SIZE parameter value of 10 on the DEFBD command, the value is *not* changed. This exception of handling FORMS_SIZE=10 will be removed in a future CDCNET release.

UNDEFINED_FE_ACTION (UNDFA)

Indicates the action the TIP should take with undefined format effectors. The following keyword values are allowed:

Keyword Value	Description
PAS	Print after spacing
PBS	Print before spacing
DPL	Discard print line

Default is PAS.

UNSUPPORTED_FE_ACTION (UNSFA)

Indicates the action the TIP should take with format effectors defined but not supported by the device. The following keyword values are allowed:

Keyword Value	Description
PAS	Print after spacing
PBS	Print before spacing
DPL	Discard print line

Default is DPL.

VFU_LOAD_OPTION (VLO)

Indicates the presence of a loadable vertical format unit (VFU) load image for the batch device, and any restrictions on changing the VFU. The following keyword values are allowed:

Keyword Value	Description
NONE	VFU load image not present or not loadable.
INIT	Specify VFU load procedure during configuration only; VFU load procedure cannot be changed by I/O station operator or user.
OPER	Default VFU load procedure can be changed by the I/O station operator.
USER	I/O station operator can change default VFU load procedure; users can change VFP for individual files.

Default for batch devices supported by the ASYNC, URI, and X.25 Asynchronous TIPs is USER.

VLO must be NONE if SIX_LPI_SEQUENCE and EIGHT_LPI_SEQUENCE are defined for the printer terminal model.

CONTROL_CODE_REPLACEMENT (CCR)

This parameter is supported by the ASYNC, X.25 Asynchronous, and URI TIPs only. It specifies a list of <control_code> and <replacement_code> pairs, that is, (<control_code>, <replacement_code>).

The <control_code> identifies a control code or range of control codes that should not be transmitted to the device. The <replacement_code>, which is optional, identifies the code to be transmitted to the device in place of the control code or codes identified by <control_code>. If <replacement_code> is not specified, no code is transmitted in place of <control_code>.

Allowed values for <control_code> are 0 through 31 and 128 through 159. Allowed values for <replacement_code> are 0 through 255.

This attribute is in effect for both transparent and nontransparent output received from the connected application. CCR processing of data not originating from the connection is done as follows:

VFU data	No CCR processing done.
INITIALIZATION_PROCEDURE INITIALIZATION_SEQUENCE FILE_PREFIX_PROCEDURE FILE_PREFIX_SEQUENCE FILE_SUFFIX_SEQUENCE	CCR processing done.
CHANNEL_1_SEQUENCE .. CHANNEL_12_SEQUENCE SINGLE_SPACE_SEQUENCE NO_SPACE_SEQUENCE FORM_FEED_SEQUENCE	NO CCR processing done.
EIGHT_LPI_SEQUENCE SIX_LPI_SEQUENCE	CCR processing is done when these sequences are output prior to a file being printed. No CCR processing is done when these sequences are sent as a result of an S or T format effector within a print file.

DATA_PARITY (DP)

Specifies parity for data received and transmitted on this line. The following values are allowed for DATA_PARITY:

Keyword Value	Description
ZERO	The parity bit is always zero.
MARK	The parity bit is always 1.
EVEN	The parity bit is set so the sum of the parity and data bits is an even value.
ODD	The parity bit is set so the sum of the parity and data bits is an odd value.
NONE	The parity bit is considered a data bit.

Default is EVEN.

This parameter is only meaningful for asynchronous and X.25 devices. For an asynchronous device to use an 8-bit code set, either Control Data-defined or site-defined, set data parity to NONE.

START (S)

Specifies whether or not the batch device should be initialized and started. Default is TRUE; START.

TRAILER_PAGE (TP)

Specifies whether or not a trailer page should be generated for the device. Default is TRUE, generate the trailer page.

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.

--ERROR-- VFU_LOAD_OPTION must be NONE if SIX_LPI_SEQUENCE and EIGHT_LPI_SEQUENCE are defined for the printer terminal model.

--FATAL-- Not enough memory currently exists for required table space.

Remarks If a standalone printer is used on NOS, it is 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 overrides 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 and if the TRAILER_PAGE parameter is not specified, no accounting message (such as "TRANSFER COMPLETE - nnnnnnn LINES PRINTED") is printed at the end of the file. In a future release of CDCNET, the BPC parameter will have no impact on whether or not the accounting message is printed.

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)
Terminal Definition Procedure Command

Purpose Defines a Network Transfer Facility (NTF) batch stream associated with the directly connected NTF remote system defined by the DEFINE_REMOTE_SYSTEM command described in this manual. 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:

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. Defaults to the name from the DEFINE_LINE command.

STREAM_ORDINAL (SO)

Specifies the stream ordinal. The stream ordinal uniquely identifies each transmit or receive stream. The NTF TIP supports up to seven receive streams for the following stream types:

PRINTER
PUNCH
PLOTTER
REMOTE_SYSTEM_INPUT
JOB_RECEIVER (with SYSOUT_RECEIVER, up to combined total of eight)
SYSOUT_RECEIVER

The NTF TIP supports up to seven transmit streams for the following stream types:

READER
JOB_TRANSMITTER (with SYSOUT_TRANSMITTER, up to a combined total of eight)
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. Default 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. Default is no file size limit is defined.

TRANSMISSION_BLOCK_SIZE (TBS)

Specifies the transmission block size to be used by the NTF TIP for communication with the remote system on the specified batch stream. For NJE remote systems, this parameter is ignored, since the transmission block size for all batch streams is negotiated during sign-on processing. For HASP remote systems, the transmission block size on the DEFINE_BATCH_STREAM command is compared to the transmission block size on the DEFINE_REMOTE_SYSTEM command, and the smaller of the two values is used.

PAGE_WIDTH (PW)

Specifies the number of columns that constitute a card image for the card reader stream to a HASP remote system. Default 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)

This parameter is currently ignored, and will be implemented in a future release. Specifies the number of cards/lines at the beginning of a HASP PUNCH, PLOT, or REMOTE_SYSTEM_INPUT stream to be discarded. HASP remote systems typically precede data on punch streams with banner or lace cards. This parameter is ignored for NJE remote systems. The default values depend on the value of the REMOTE_SYSTEM_TYPE parameter on the DEFINE_REMOTE_SYSTEM command, and are as follows:

REMOTE_SYSTEM_TYPE	Default Skip Punch Count
NOS_VE	0
NOS	2
NOS_BE	1
IBM	0
DEC	0
USER	0

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 is TRUE.

Responses

Batch stream <name> 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 an 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) Terminal Definition Procedure Command

Purpose

Defines auto-configured private and public I/O stations on NOS and NOS/VE. It can also be used to define operator-configured public I/O stations on NOS and NOS/VE. Do not use this command to configure an operator-configured private I/O station; instead, use DEFINE_USER_I_O_STATION or DEFINE_NP_BATCH_STATION. 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, and operator-configured public 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 these 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. This bulletin provides installation information. The ACTIVATE_SCFS command is described in the NOS/VE Network Management manual.

NOTE

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.

If the I/O station connects 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/VE and NOS.

DEFAULT_JOB_DESTINATION (DJD)

Specifies the destination to which an input file is 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. If this parameter is omitted and no ROUTE_JOB destination is specified, the DESTINATION_UNAVAILABLE_ACTION is taken.

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.

This parameter must be specified if the I/O station is to connect to NOS RBF. This parameter must *not* be specified if the I/O station is to connect to NOS PSU.

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 (DUA)

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 is read and discarded and the reading of subsequent jobs continues if the destination is unavailable.
STOP	The input device for the job is stopped and no more jobs are 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 or 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-- Unable to start task <entry_point_name>.

<Reason required task cannot be started>

--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=display,..
  i_o_station_alias=(rem1,engbldg)
```

DEFINE_NP_BATCH_STATION (DEFNBS) (NOS Only) Terminal Definition Procedure Command

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) that supports the station.

Format **DEFINE_NP_BATCH_STATION**

Parameters None.

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-- Unable to start task <entry_point_name>. <Reason required task cannot be started>

--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 four-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 with system ID of 0800251FE029 that has last assigned number 1234 is named $IOSTATION_1FE029_1235.

Examples

```
define_np_batch_station
```

DEFINE_REMOTE_SYSTEM (DEFRS) Terminal Definition Procedure Command

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
DEFAULT_JOB_DESTINATION = name
DEFAULT_FILE_DESTINATION = name
LOGICAL_LINE_NUMBER = 1..999
REMOTE_SYSTEM_PROTOCOL = keyword value
REMOTE_SYSTEM_TYPE = keyword value
ROUTE_BACK_POSITION = 0..254 or *keyword value*
LINE_NAME = name
TERMINAL_USER_PROCEDURE = name
AUTHORITY_LEVEL = keyword value
POSITIVE_ACKNOWLEDGE = keyword value
WAIT_A_BIT = keyword value
INACTIVITY_TIMER = 1..600 or *INFINITE*
RECEIVE_PASSWORD = string 1..8
TRANSMIT_PASSWORD = string 1..8
TRANSMISSION_BLOCK_SIZE = 400..4095
REMOTE_IDENTIFIER = string 1..3
SIGNON = string 1..80

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 sign-on 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".

DEFAULT_JOB_DESTINATION (DJD)

Specifies the destination to which all NJE received jobs are sent. For HASP remote systems, this parameter specifies the destination to which received jobs are sent if no destination is specified on the ROUTE_JOB command for the job. 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 all NJE received files are sent. For HASP remote systems, this parameter specifies the destination to which received files are sent if no destination is specified on the ROUTE_FILE command for the job. A file destination is a family name registered (as BTFS$family_name) by a CYBER 180 NOS/VE system.

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; this allows remote operators to reference streams on specific lines.

REMOTE_SYSTEM_PROTOCOL (RSP)

Specifies the protocol used to communicate with the remote system. The following keyword values are allowed:

NJE
HASP

Either NJE or HASP must be specified.

REMOTE_SYSTEM_TYPE (RST)

This parameter may be specified for both NJE and HASP remote systems. The following keyword values are allowed:

NOS_VE
NOS
NOS_BE
IBM
DEC
USER

Default is IBM.

If the remote system supports the HASP protocol, the REMOTE_SYSTEM_TYPE parameter value is used to determine the routeback information format to insert when sending jobs to the remote system. Routeback is the mechanism by which NTF propagates file ownership, destination, and routing characteristics.

ROUTE_BACK_POSITION (RBP)

Specifies the position in the job after which routeback information is inserted by NTF. Specifying NONE indicates no routeback information is inserted. Specifying 0 indicates routeback information is inserted as the first record of the job. This parameter is ignored for NJE remote systems. Defaults depend on the value of the REMOTE_SYSTEM_TYPE parameter, and are as follows:

REMOTE_SYSTEM_TYPE	Default Routeback Position
NOS_VE	3
NOS	3
NOS_BE	3
IBM	2
DEC	2
USER	NONE

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. Default is the name from the DEFINE_LINE command.

TERMINAL_USER_PROCEDURE (TUP)

This parameter is currently ignored, and will be implemented in a future release. 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. Default is no TUP is executed. This parameter is for HASP remote systems, and is ignored for NJE remote systems.

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.

This parameter is used for NJE remote systems, but is ignored for HASP remote systems.

POSITIVE_ACKNOWLEDGE (PA)

Specifies what sequence should be sent to the remote system as a positive acknowledgement, 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 sign-on processing, ACK is the only valid response following receipt of 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. Default 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.

Default is ACK.

If NULL is specified for the POSITIVE_ACKNOWLEDGE parameter, then FCS is the only value that 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 (in 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 INFINITE disables the timer and causes the DI to wait indefinitely for a response. Default is INFINITE. This parameter is for HASP remote systems, and is ignored for NJE remote systems.

RECEIVE_PASSWORD (RP)

Specifies the node password to be received from the NJE remote system during sign-on processing. Default is no password. This parameter is ignored for HASP remote systems.

TRANSMIT_PASSWORD (TP)

Specifies the password to be sent to the HASP or NJE remote system during sign-on processing. For an NJE remote system, this value is the node password used during sign-on processing. Default is no password.

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.

REMOTE_IDENTIFIER (RI)

Specifies the remote identifier to be used on the "standard format" sign-on for a HASP remote system. Default for HASP remote systems is 1. This parameter is ignored for NJE remote systems.

SIGNON (S)

Allows the specification of a custom or "nonstandard format" sign-on for a HASP remote system. Default is a "standard format" sign-on sent using REMOTE_IDENTIFIER and TRANSMIT_PASSWORD parameter values. This parameter is ignored for NJE remote systems.

Responses Remote System <name> 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-- Unable to start task <entry_point_name>. <Reason required task cannot be started>

--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, ..
  default_job_destination=CLSH109, ..
  default_file_destination=CLSH109
```

DEFINE_TERMINAL_DEVICE (DEFTD) Terminal Definition Procedure Command

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, Control Data-provided TIPs only support the definition of CONSOLE devices via the DEFINE_TERMINAL_DEVICE command. This command may be executed only by inclusion in a TDP.

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.

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
VALIDATE_USERS = boolean

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. For more information, see the CDCNET Configuration Guide. 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 six 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_4200000000.

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)

Specifies the cluster address to be used by the TIP for communication with the batch device. This parameter is used for BSC3270 and MODE4 TIPs. Default is 0.

For the HASP TIP, only one cluster is allowed on each line so this parameter is not used.

For the MODE4 TIP, the CLUSTER_ADDRESS must be in the range of 70(16) through 7F(16) for Mode 4A and 20(16) through 7F(16) for Mode 4C clusters. Since the default is 0, this parameter must be specified for all MODE4 devices.

For the BSC3270 TIP, the cluster address must be in the range of 0 through 31. Default is 0.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the cluster address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

DEVICE_ADDRESS (DA)

Specifies the device address to be used by the TIP for communication with the console device. This parameter is used for HASP, BSC3270 and MODE4 TIPs. For devices supported by the HASP TIP, the DEVICE_ADDRESS parameter is ignored if the DEVICE_TYPE=CONSOLE, and need not be specified. Default is 0.

For the MODE4 TIP, the DEVICE_ADDRESS must be 61(16) for all Mode 4A devices and in the range of 61(16) through 6F(16) for Mode 4C devices. Since the default is 0, this parameter must be specified for all MODE 4 devices.

For TIPS which support the X.PC protocol, if TDPs are used to configure the terminal devices (see DEFINE_LINE command in this manual), the TDP must not contain a DEFTD command with a nonzero device address. The X.PC protocol starts only when the device address is set to or defaults to zero.

For the BSC3270 TIP, the device address must be in the range of 0 through 31. Default is 0.

NOTE

In the configuration sequence that is entered at the 3270 workstation, the device address must be specified using a different value from the one specified on this command. See the CDCNET Configuration Guide for more information.

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
NJEF	None
HASP	CONSOLE
MODE4	CONSOLE
NTF	None
NTF	None
SNA3270	CONSOLE
URI	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 (described in this manual) for this device only. Default 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).

VALIDATE_USERS (VU)

Specifies if users on this device are validated when user validation is enabled by the DEFINE_USER_VALIDATION command. If TRUE is specified, users are validated when validation is enabled. If FALSE is specified, users do not need to be validated.

The value on this command overrides the VU parameter on the DEFINE_LINE or DEFINE_TIP command on this device only. Default is the value specified on the DEFINE_LINE command or on the DEFINE_TIP command if VU is not specified on the DEFL 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) Terminal Definition Procedure Command

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.

NOTE

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 is 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. If this parameter is omitted and no ROUTE_JOB destination is specified, the DESTINATION_UNAVAILABLE_ACTION is taken.

DESTINATION_UNAVAILABLE_ACTION (DUA)

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 is read and discarded and the reading of subsequent jobs continues if the destination is unavailable.
STOP	The input device for the job is stopped and no more jobs are 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-- Unable to start task <entry_point_name>. <Reason required task cannot be started>

--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 four-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 with a system ID of 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
```

This chapter contains descriptions of commands that can be used only in terminal user procedures (TUPs). Currently there are three such commands: The following commands (documented in the CDCNET Terminal Interface 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)
Terminal User Procedure Command

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
DATA_MODE = 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 default is TERMINAL. 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.

DATA_MODE (DM)

Specifies the input editing mode to be used when sending the message STRING. This parameter applies only for a DESTINATION of CONNECTION. The default is NORMAL. Transparent mode is useful for Passthrough and X.25 Terminal Gateway client connections to instruct the server to forward the data to the remote destination.

Keyword Value	Description
NORMAL (N)	VTP normal mode
TRANSPARENT (T)	VTP transparent mode

Remarks

- You cannot use the network command character (% by default) as the first character in a PUT_STRING command. However, the string value on PUT_STRING can contain the network command character.
- 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.
- 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
```

NOTE

PUT_STRING commands can be used to log into a host and execute an initial set of commands. However, some hosts require a delay between the login statement and subsequent commands. For delay processing information, see the WAIT command later in this chapter.

SET_PAD_MESSAGE (SETPM)
Terminal User Procedure Command

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 (TUP).

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 15-1 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. Therefore, you should not use the SETPM command to set X.3 parameters that are mapped to VTP attributes, because results might be unpredictable.

Table 15-2 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 a terminal user, an 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) is set to 0 (enabled).

```
SETPM,value=((3,2),(0,33),(63,0))
```

Table 15-1. 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. No data forwarding on timeout.
5	Ancillary device control	Decimal 0. No use of X-ON(DC1) and X-OFF(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). 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. Normal data delivery to the start-stop mode DTE.
9	Padding after carriage return (CR)	Decimal 0. The PAD never performs padding after a carriage return.
10	Line folding	Decimal 0. The PAD never performs 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 15-1. 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. No use of X-ON(DC1) and X-OFF(OFF).
13[2]	Line feed insertion after carriage return	Decimal 4. 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. 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 15-2. PAD Parameter/CDCNET Attribute Correlation

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.

(Continued)

Table 15-2. PAD Parameter/CDCNET Attribute Correlation *(Continued)*

PAD Reference	CDCNET Attribute(s)	Effect on Setting
3		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.
4	IEM	See description of 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, then: 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.

WAIT
Terminal Definition Procedure or Terminal User Procedure Command

Purpose Delays execution of the next command(s) in a procedure file. This command may originate only from a terminal user procedure or a terminal definition procedure.

For example, you can use WAIT when sending PUT_STRING data to a host application that has timing constraints. After a PUT_STRING logs a user into RBF and the console has been logged in, RBF requests the batch devices. When the next PUT_STRING references a batch device, execution of the next command must be delayed for a few seconds, to ensure that the batch device was allocated to RBF.

Format

WAIT
TIME = integer 1..100

Parameters **TIME (T)**

Specifies, in seconds, the time to delay processing of the following command in the procedure file.

Responses None defined.

Examples The following command sequence shows how to use the WAIT command with the PUT_STRING command for delay processing:

```
put_string string='',HGC,XXXXXX,IAF''..
  destination=connection
wait time=5 "Allow time for LOGIN to complete."
put_string string='MAIL''..
  destination=connection
```

WAIT (WAIT)
Terminal Definition Procedure or Terminal User Procedure Command

Purpose Delays execution of the next command(s) in a procedure file. This command may originate only from a terminal user procedure or a terminal definition procedure.

For example, you can use WAIT when sending PUT_STRING data to a host application that has timing constraints. After a PUT_STRING logs a user into RBF and the console has been logged in, RBF requests the batch devices. When the next PUT_STRING references a batch device, execution of the next command must be delayed for a few seconds, to ensure that the batch device was allocated to RBF.

Format

WAIT
TIME = integer 1..100

Parameters

TIME (T)

Specifies, in seconds, the time to delay processing of the following command in the procedure file.

Responses None defined.

Examples The following command sequence shows how to use the WAIT command with the PUT_STRING command for delay processing:

```
put_string string=',HGC,XXXXXX,IAF'..
  destination=connection
wait time=5 "Allow time for LOGIN to complete."
put_string string='MAIL'..

  destination=connection
```

Load Procedure Commands 16

Load Procedure Commands 16

This chapter contains descriptions of commands that can only be used in load procedures. Command descriptions are organized alphbetically by full command name.

Unless specified otherwise, all commands are valid in both NOS and NOS/VE operations environments.

DEFINE_CODE_SET_BASE (DEFCSB) Load Procedure Command

Purpose Specifies the name of a CDCNET standard code set against which changes specified by the TRANSLATIONS parameter of the DEFINE_CODE_SET_TRANSLATION command (described next) are to be applied. This command is optional and can occur only in a code set procedure. If used, it may only appear once and then only as the first command of the load procedure. When this command is not present in a code set procedure, the base code set used is ASCII256.

Format

DEFINE_CODE_SET_BASE
CODE_SET = keyword value

Parameters *CODE_SET (CS)*

Specifies a CDCNET-supplied code set as the base against which the translations specified in the DEFCST commands are applied. The following keyword values are allowed:

ASCII48
ASCII64
ASCII95
ASCII128
ASCII256
EBCDIC

Default is ASCII256.

Responses Code set base defined.

--ERROR-- Command DEFINE_CODE_SET_BASE is not allowed in proc type <xxx>.

--ERROR-- DEFCSB is not the first command or it appears more than once in the load procedure.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_code_set_base ..
  code_set = EBCDIC
```

DEFINE_CODE_SET_TRANSLATION (DEFCST) Load Procedure Command

Purpose Defines a site-specific code set. This command is only valid within a code set procedure. The name of the procedure containing the DEFCST commands is used as the name of the code set being defined. A code set procedure can contain multiple DEFCST commands, their cumulative effect being the definition of a site-defined code set. A code set procedure defines one and only one code set.

Format

DEFINE_CODE_SET_TRANSLATION or
DEFINE_CODE_SET_TRANSLATIONS
TRANSLATIONS = list 1..64, 2..2 of <ccode>

Parameters

TRANSLATIONS (T)

Specifies a list of up to 64 individual code translation pairs. Each pair consists of a <code> and a <translation code>, that is, (<code>,<translation_code>).

For output devices, <code> specifies the network's code, and <translation_code> specifies the code that is to be sent to the device in place of the network's code. Network codes which do not appear in any individual code translation pair are translated according to the base code set when they appear in output received from network applications.

For input devices, <code> specifies the device's code and <translation_code> specifies the network code that is to be sent to the network application in place of the device's code.

If a particular value appears as the <code> in more than one translation pair in one or more DEFCST commands in the same procedure, it is considered multiply defined. In this case, the last definition takes effect when the actual code set is constructed.

Responses Code set translations defined.

--ERROR-- Command DEFINE_CODE_SET_TRANSLATION is not allowed in proc type <xxx>.

--FATAL-- Not enough memory is currently available for required table space.

Examples

```
define_code_set_translations ..
  translations = ((4E(16),26(16)),(50(16),2B(16)))
```

DEFINE_VFU_LOAD_IMAGE (DEFVLI) Load Procedure Command

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 concatenates the supplied procedure name with the string LOAD_PROCEDURE# and accesses 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 six and eight lines per inch (suitable for printers with a terminal model value of CDC_585V).

If you do not specify a default VFU load image for a printer that has a loadable VFU, the DI supporting the printer uses the Control Data-provided VFU load procedure (CDC_VFU) for user-defined terminal models. For Control Data-defined models, the default is VFU_model_name. 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.

See 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..127 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 that are set for the line.

Responses VFU line/channel pair defined.

--ERROR-- Command DEFINE_VFU_LOAD_IMAGE is not allowed in proc type <xxx>.

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. Lines not specified are filled in by the command processor with zeros, up to the page length specified by the file.

For URI TIP-supported printers, and for CDC_537V, the following apply:

- 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 earlier in this chapter) are not specified in the load procedure, then the bottom-of-form channel is provided at the line number with the value <page length minus 2>, and the auto page-eject channel is provided at the line after that.
- If one of those channels is specified in the load procedure, the other is defined so that the auto page-eject channel is in the line following the bottom-of-form channel.

For other printer models, the auto page-eject channel and the bottom-of form channel are set in the same line.

If an operator or user changes FORMS_SIZE or VERTICAL_PRINT_DENSITY batch device attributes (using the CHANGE_BATCH_DEVICE_ATTRIBUTES command or the T or S format effectors), the bottom-of-form and auto page-eject channels are moved to accommodate the change in number of lines on the form. If the auto page-eject option is changed, the auto page-eject channel is removed or restored (depending on 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 is sent to the printer's loadable VFU.

Top-of-form channel (channel 1), bottom-of-form channel, and auto page-eject channel may each be specified only once in a load procedure. Auto page-eject channel may not be defined for line number 1.

Examples These examples of DEFINE_VFU_LOAD_IMAGE are from released VFU load procedure CDC_VFU, which 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
```

PUT_BATCH_SEQUENCE (PUTBS)
Load Procedure Command

Purpose Specifies data that should be output to a batch device. For example, if this command appears in a file prefix procedure, it specifies data to be sent to a printer before each nontransparent file is printed. This command may be executed only from a load procedure.

Format

PUT_BATCH_SEQUENCE
SEQUENCE = list 1..31 of string or <ccode>

Parameters

SEQUENCE (S)

Specifies a sequence of control codes or characters of data to be sent to a batch output device.

NOTE

The following character strings have special significance for the POSTSCRIPT printer model and site-defined printer models starting with 'POSTSCRIPT_'. They should not be used within data in START_POSTSCRIPT_FILE or in other FPPs, because they may have special uses for other printer models in a future release.

$PW

$FS

Responses

--ERROR-- Command PUT_BATCH_SEQUENCE is not allowed in proc type <xxx>.

--ERROR-- A sequence parameter must be of type ccode or string.

Examples

```
put_batch_sequence sequence=(CR,LF,'STRING1',0A(16))
```

Configuration File Management Commands 17

Configuration File Management Commands 17

This chapter describes commands that access CDCNET configuration procedures on NOS/VE and NOS hosts. Use these procedures after creating configuration procedures, and before/after editing configuration procedures. On NOS/VE, these procedures simplify file access. To make these procedures available NOS/VE, you must enter the following:

```
$system.CDCNET.version_independent.command_library
```

On NOS, these procedures access the Network File Manager Utility (NETFM) and Network Procedure Library Manager (NETPLM) to change the network directory and update configuration procedure libraries and files.

GET_CDCNET_PROCEDURE (GETCP) (NOS/VE Only) NETOU Command

Purpose Gets a configuration file or procedure from a configuration procedure library in the $SYSTEM.CDCNET.SITE_CONTROLLED catalog. This command has the alias ATTACH_CONFIGURATION_FILE.

Format

GET_CDCNET_PROCEDURE or
ATTACH_CONFIGURATION_FILE
FILE = file
FILE_TYPE = keyword value or **PROCEDURE_TYPE**
PROCEDURE_NAME = keyword value
STATUS = status variable

Parameters

FILE (F)

The name of the local file that receives the contents of the configuration file or procedure when it is attached.

FILE_TYPE (FT) or **PROCEDURE_TYPE (PT)**

Indicates the type of procedure file to be attached. The following keyword values are allowed:

SYSTEM_CONFIGURATION (SC)
TERMINAL_USER_PROCEDURE (TUP)
TERMINAL_DEFINITION_PROCEDURE (TDP)
LOAD_PROCEDURE (LP) (alias for VFU_LOAD_PROCEDURE)
EXCEPTION_LIST (ELIST) or (EL)

PROCEDURE_NAME (PN)

The name of the procedure within the appropriate procedure library that is to be attached. This parameter is required for procedure types SYSTEM_CONFIGURATION, TERMINAL_USER_PROCEDURE, TERMINAL_DEFINITION_PROCEDURE, and LOAD_PROCEDURE. It does not apply to and is not required for file type EXCEPTION_LIST.

For DI system configuration procedures (procedure type SC), specify the last six digits of the DI's system ID, and include the radix (16) (see example). For procedure types TUP, TDP, and LP, specify the entire name of the TUP, TDP, or LP.

By specifying the keyword value ALL (PN=ALL), all procedures from the specified library are written to the specified file.

STATUS

SCL status variable. See the NOS/VE System Usage manual for information on this parameter.

Examples This example attaches the configuration procedure for a DI with a system ID of 080025300117 to a local file named $LOCAL.B:

```
getcp $local.b sc 300117(16)
```

REPLACE_CDCNET_PROCEDURE (REPCP) (NOS/VE Only) NETOU Command

Purpose Replaces a configuration procedure or file in the appropriate procedure library in the $SYSTEM.CDCNET.SITE_CONTROLLED catalog. This command has the aliases DEFINE_CONFIGURATION_FILE (DEFCF) and REPLACE_CONFIGURATION_FILE (REPCF).

Format

REPLACE_CDCNET_PROCEDURE or
DEFINE_CONFIGURATION_FILE or
REPLACE_CONFIGURATION_FILE
FILE = **file**
FILE_TYPE = **keyword value** or **PROCEDURE_TYPE**
STATUS = status variable

Parameters

FILE (F)

The name of the local file that contains the configuration file or procedure to be placed in the procedure library.

FILE_TYPE (FT) or **PROCEDURE_TYPE (PT)**

Indicates the type of file or procedure library in the $CDCNET.SYSTEM.SITE_CONTROLLED catalog that is updated with the contents of the local file. The following keyword values are allowed:

SYSTEM_CONFIGURATION (SC)
TERMINAL_USER_PROCEDURE (TUP)
TERMINAL_DEFINITION_PROCEDURE (TDP)
LOAD_PROCEDURE (LP) (alias for VFU_LOAD_PROCEDURE [VLP])
EXCEPTION_LIST (ELIST) or (EL)

STATUS

SCL status variable. See the NOS/VE System Usage manual for information on this parameter.

Examples In this example, the file $LOCAL.B that was used in the GET_CDCNET_PROCEDURE example has been edited. Now, the SC procedure library must be updated. The following command is used to update the SC procedure library:

```
repcp $local.b sc
```

SET_VERSION_LEVEL (SETVL) (NOS/VE Only) NETOU Command

Purpose Changes the default version level of software to be loaded into each DI and the default version level of the three CDCNET host utilities (ANACD, MANCC, and NPA).

Format

SET_VERSION_LEVEL
PROCESSING_OPTION = list of keywords
VERSION = level
STATUS = status variable

Parameters

PROCESSING_OPTION (PO)

Specifies which defaults are being set. The following keywords can be specified:

Keyword	Description
UTILITIES (U)	Sets the default version for the three host utilities. The file $SYSTEM.CDCNET.VERSION_INDEPENDENT.CDCNET_VERSION is updated.
EXCEPTION_LIST (EL)	Sets the default version for software to be loaded into each DI. The file $SYSTEM.CDCNET.SITE_CONTROLLED.EXCEPTION_LIST is updated.
ALL	Sets the default version for both UTILITITES and EXCEPTION_LIST files.

VERSION (V)

A 4-digit hexadecimal version level.

STATUS (S)

SCL status variable. See the NOS/VE System Usage manual for information on this parameter.

Remarks You must have ALL, CYCLE, and CONTROL privileges for the specific files or catalogs containing the files you wish to update.

Do not use this command to update the exception list if you are managing your CDCNET configuration with NETCU. The NETCU configuration database is not updated by this command.

When you update the exception list, the command searches for the string dv=vvvv(16). If this string is not in the file, the command does not update the exception list.

SETVL does not change the level specified on any DEFINE_EXCEPTION_SYSTEM command.

Examples This example sets the default version for both UTILITITES and EXCEPTION_LIST files to 7305.

```
setvl,po=all,v=7305
```

GETCP (Get CDCNET Procedure) (NOS MDI Only) NETOU Command

Purpose Gets the specified procedure from the appropriate CDCNET procedure library (SCLIB, TDPLIB, TUPLIB, or LPLIB), or gets the exception list using NETFM and writes the procedure or exception list in the specified file.

Format **BEGIN,GETCP,DCNPLIB,FILE,PT,PN,*LIBRARY***

Parameters

FILE (F)

Specifies the name of the local file that receives the contents of the configuration procedure when it is attached.

PT

Specifies the procedure type. Allowed values include:

Value	Description
EL	Exception list
SC	System configuration procedures
TDP	Terminal definition procedures
TUP	Terminal user procedures
LP	Load procedures

PN

Specifies the procedure name. The PN value relates to the PT value, as shown in the following table.

PT	PN Value
EL	There is no variable portion for exception lists. Do not enter a value.
SC	The last six digits of the DI's system ID. GETCP provides the first characters (SYSTEM_080025) for you.
TDP	The complete TDP name, which must match the name on the PROC statement in the TDP.
TUP	The complete TUP name, which must match the name on the PROC statement in the TUP.
LP	The complete load procedure name, which must match the name on the PROC statement in the load procedure.
ALL	Specifies that all procedures in a library are written to the file specified by the FILE parameter.

LIBRARY

Specifies the procedure library name. You need specify this parameter only to get a procedure in a library not listed in the CDCNET directory. This parameter specifies the NOS direct access permanent file on the current user name containing the procedure library.

Remarks When using GETCP in line mode, if a procedure name contains special characters, particularly commercial at (@), you must bracket the procedure name by dollar signs ($). For example, procedure @PSUTUP is specified as follows:

```
begin,getcp,dcnplib,uppsu,tup,$@psutup$
```

The dollar signs are not required when using GETCP in screen mode.

Examples The following example gets the system configuration procedure SYSTEM_080025100021 from procedure library SCLIB, and copies it to local file C100021. Since the variable portion of a system configuration procedure is the last six digits of the system ID, 100021 is specified for the PN (procedure name) parameter.

```
begin,getcp,dcnplib,c100021,sc,100021
```

The following example gets terminal definition procedure RBF_WORK_STATION_1 from procedure library TDPLIB, and copies it to local file TPRBFW1. Since the variable portion of a terminal definition procedure is the entire procedure name, RBF_WORK_STATION_1 is specified for the PN (procedure name) parameter.

```
begin,getcp,dcnplib,tprbfw1,tdp,rbf_work_station_1
```

REPCP (Replace CDCNET Procedure) (NOS MDI Only) NETOU Command

Purpose Places the specified file in the appropriate CDCNET procedure library (ELIST, SCLIB, TDPLIB, TUPLIB, or LPLIB).

Format BEGIN,REPCP,DCNPLIB,FILE,PT,*LIBRARY*

Parameters

FILE (F)

Specifies the name of the local file containing the procedure(s) to be replaced.

PT

Specifies the procedure type. Allowed values include:

Value	Description
EL	Exception list
SC	System configuration procedures
TDP	Terminal definition procedures
TUP	Terminal user procedures
LP	Load procedures

LIBRARY

Specifies the procedure library name. You need specify this parameter only to replace a procedure in a library not listed in the CDCNET directory. This parameter specifies the NOS direct access permanent file on the current user name containing the procedure library.

Examples The following example replaces the system configuration procedure in local file C100021 in procedure library SCLIB.

```
begin,repcp,dcnplib,c100021,sc
```

The following example replaces the TUP in local file UPPSU, in procedure library TUPLIB.

```
begin,repcp,dcnplib,uppsu,tup
```

The following example replaces the TDP in local file TPRBFW1, in procedure library TDPLIB.

```
begin,repcp,dcnplib,tprbfw1,tdp
```

ATTCF (Attach Configuration File) (NOS MDI Only) NETOU Command

Purpose Attaches the specified file from the network directory, and converts the file from ASCII88 character code set to ASCII 6/12. Use this procedure before you edit a configuration procedure or exception list, and before a MANCC USE_CONFIGURATION_FILE subcommand.

Format **BEGIN,ATTCF,DCNPLIB,LFN,PFN,TYPE,NETNAME**

Parameters

LFN

The local ASCII 6/12 file to receive a copy of the configuration procedure.

PFN

The ASCII88 permanent file to get, save, or replace. If lfn and pfn have the same value, enter an asterisk (*) for this parameter.

TYPE

Specifies the procedure type. Allowed values include: EL (exception list), SC (system configuration file), TDP (terminal definition procedure), TUP (terminal user procedure), and LP (load procedure). This parameter generates the fixed portion of the network file name (NFN) for the file, which is assigned to each file type as follows:

Type	Fixed Portion of NFN
EL	EXCEPTION_LIST
SC	CONFIGURATION#080025
TDP	TERMINAL_PROCEDURE#
TUP	USER_PROCEDURE#
LP	LOAD_PROCEDURE#

The fixed portion of the NFN is combined with the variable portion of the file name (see NETNAME parameter) to generate a complete NFN.

NETNAME

Specifies the variable portion of the NFN. The netname value is combined with the fixed portion (see type parameter) to generate a complete NFN. The netname parameter relates to the type parameter as follows:

Type	Netname Value
EL	Not needed. Use value 0 for DEFCF.
SC	Last six digits of the DI's system ID.
TDP	The TDP name, which must match the name on the PROC statement in the TDP.
TUP	The TUP name, which must match the name on the PROC statement in the TUP.
LP	The load procedure name, which must match the name on the PROC statement in the load procedure.

Examples In this example, file c100022 is attached and converted from ASCII88 to ASCII 6/12. File c100021 receives a copy of the configuration procedure.

```
begin,attcf,dcnplib,c100022,c100021
```

DEFCF (Define Configuration File) (NOS MDI Only) NETOU Command

Purpose Defines a configuration procedure or file in the network directory for the first time. DEFCF converts the specified file to ASCII88 character code set from ASCII 6/12, and saves the converted file. DEFCF also creates an entry in the CDCNET file directory. Use this procedure after you have created a CDCNET configuration procedure or have used the MANCC CREATE_CONFIGURATION_FILE (CRECF) subcommand.

Format **BEGIN,DEFCF,DCNPLIB,LFN,PFN,TYPE,NETNAME**

Parameters

LFN

The local ASCII 6/12 file to receive a copy of the configuration procedure.

PFN

The ASCII88 permanent file to get, save, or replace. If lfn and pfn have the same value, enter an asterisk (*) for this parameter.

TYPE

Specifies the procedure type. Allowed values include: EL (exception list), SC (system configuration file), TDP (terminal definition procedure), TUP (terminal user procedure), and LP (load procedure). This parameter generates the fixed portion of the network file name (NFN) for the file, which is assigned to each file type as follows:

Type	Fixed Portion of NFN
EL	EXCEPTION_LIST
SC	CONFIGURATION#080025
TDP	TERMINAL_PROCEDURE#
TUP	USER_PROCEDURE#
LP	LOAD_PROCEDURE#

The fixed portion of the NFN is combined with the variable portion of the file name (see NETNAME parameter) to generate a complete NFN.

NETNAME

Specifies the variable portion of the NFN. The netname value is combined with the fixed portion (see type parameter) to generate a complete NFN. The netname parameter relates to the type parameter as follows:

Type	Netname Value
EL	Not needed. Use value 0 for DEFCF.
SC	Last six digits of the DI's system ID.
TDP	The TDP name, which must match the name on the PROC statement in the TDP.
TUP	The TUP name, which must match the name on the PROC statement in the TUP.
LP	The load procedure name, which must match the name on the PROC statement in the load procedure.

Examples In this example, file c100000 is converted from ASCII 6/12 to ASCII88 and placed in file c100001. The configuration procedure being defined is an SC (system configuration file).

```
befin,defcf,dcnplib,c100001,c100000,sc,080025300011
```

REPCF (Replace Configuration File) (NOS MDI Only) NETOU Command

Purpose Replaces the existing configuration procedure or file (same network file name and permanent file name) in the network directory. REPCF converts the specified file to ASCII88 character code set from ASCII 6/12, replaces the file in the NOS permanent file catalog, and updates the directory entry for the file in the CDCNET directory file. Use this procedure after you have edited a configuration procedure using a file editor or the MANCC EDIT_CONFIGURATION_FILE (EDICF) subcommand.

Format **BEGIN,REPCF,DCNPLIB,LFN,PFN,TYPE,NETNAME**

Parameters

LFN

Specifies the local ASCII 6/12 file to receive a copy of the configuration procedure.

PFN

Specifies the ASCII88 permanent file to get, save, or replace. If lfn and pfn have the same value, enter an asterisk (*) for this parameter.

TYPE

Specifies the procedure type. Allowed values include: EL (exception list), SC (system configuration file), TDP (terminal definition procedure), TUP (terminal user procedure), and LP (load procedure). This parameter generates the fixed portion of the network file name (NFN) for the file, which is assigned to each file type as follows:

Type	Fixed Portion of NFN
EL	EXCEPTION_LIST
SC	CONFIGURATION#080025
TDP	TERMINAL_PROCEDURE#
TUP	USER_PROCEDURE#
LP	LOAD_PROCEDURE#

The fixed portion of the NFN is combined with the variable portion of the file name (see NETNAME parameter) to generate a complete NFN.

NETNAME

Specifies the variable portion of the NFN. The netname value is combined with the fixed portion (see type parameter) to generate a complete NFN. The netname parameter relates to the type parameter as follows:

Type	Netname Value
EL	Not needed. Use value 0 for DEFCF.
SC	Last six digits of the DI's system ID.
TDP	The TDP name, which must match the name on the PROC statement in the TDP.
TUP	The TUP name, which must match the name on the PROC statement in the TUP.
LP	The load procedure name, which must match the name on the PROC statement in the load procedure.

CAUTION

These procedures display NETFM output as they access the CDCNET network directory file, NETDIR. Always check this output to ensure that the procedure completed successfully, particularly after the DEFCF and REPCF procedures.

These procedures are in file DCNPLIB. If you need to display the ATTCF, DEFCF, and REPCF procedures themselves, Control Data recommends that you do not use Full Screen Editor (FSE) to view them. Changes may occur to files that contain the @ character.

Examples

In this example, the existing configuration file c111111 is converted from ASCII 6/12 to ASCII88 and returned to the network directory. Local file c121212 receives a copy of the configuration file.

```
begin,repcf,dcnplib,c121212,c111111
```

SETVL (Set Version Level Procedure) (NOS MDI Only) NETOU Command

Purpose

When a new version of Batch Corrective Update (BCU) for CDCNET is installed, the version level changes. The SETVL procedure updates the version level.

The version level is included in all released CDCNET software files except site-controlled configuration files. Version levels in the file names allow multiple versions of CDCNET to reside together on disk under one user name.

NOTE

If you installed CDCNET for MDI Reset Support Only, there is no need to activate the new level of CDCNET because there is no CDCNET software to load.

The version level of the components of CDCNET are controlled in two places: in the INITDCN procedure (stored in file INITDCN on the network administration user name) and in the exception list (file ELIST on the network administration user name). The exception list maintains the default version level for all non-host-connected DIs; INITDCN maintains the levels for all other components.

Format

BEGIN,SETVL,DCNPLIB,TOOL=tool,VERSION=vvvv, *ID=id.*

Parameters

TOOL

Name of the component to update. Allowable values are:

ANACD
ELIST
INMD
MANCC
NETOU
NPA
ALL

NOTE

INMD refers to the boot file level INITMDI loads into all host-connected DIs, ELIST refers to the exception list, and ALL updates all the tools.

It is recommended that TOOL=ALL be used for an upgrade or BCU installation. Once SETVL has completed, all future load requests for DIs and all accesses for ANACD, MANCC, and NPA use vvvv level software; the next initiation of NETOU uses template file TF_id_vvvv.

VERSION (V)

Version level of CDCNET you wish to activate.

ID

The 1-character identifier for the CHA1 template file. The id character for the release template is A. This parameter is required for TOOL=NETOU and TOOL=ALL.

NOTE

For this procedure to properly process the exception list, the permanent file name must be ELIST, and the DEFBD (DEFINE_BOOT_DEFAULTS) command must specify the software version level with the text DV = vvvv(16). DV=vvvv(16) must be on one line. SETVL does not set the boot default for specific DIs that are handled in the exception list.

Examples

```
BEGIN,SETVL,DCNPLIB,TOOL=all,VERSION=7305,ID=a.
```

Network Performance Analyzer (NPA) Commands 18

Network Performance Analyzer (NPA) Commands 18

This chapter describes the commands and parameters used by NPA. NPA uses two major software components and four minor software components. The commands associated with the two major components are:

Data Reformatting Command	REFORMAT_CDCNET_LOG_FILE (REFCLF)
Report Generator Command	CREATE_CDCNET_ANALYSIS_REPORT (CRECAR)

NPA provides you with four other components that allow you to maintain your database files, obtain helpful information regarding processing, change expected operating limits, and automatically execute procedures periodically. The commands associated with these four components are:

File Maintenance Utilities Commands	ARCHIVE_NPA_DATA_BASE (ARCNDB) RELOAD_NPA_DATA_BASE (RELNDB)
Help File Utilities Commands	EXPLAIN_CDCNET_LOG_MESSAGE (EXPCLM) EDIT_CDCNET_LOG_MESSAGE (EDICLM)
Change Expected Operating Limits Utility Command	CHANGE_EXPECTED_OPERATING_LIMIT (CHAEOL)
Periodic Utility Command	SUBMIT BATCH JOB PERIODICALLY (SUBBJP) (NOS only)

NOTE

The command syntax descriptions used in this chapter are oriented toward the interactive user. Batch users on NOS must add the appropriate command statement terminators.

Accessing NPA

Access to NPA on NOS is accomplished by entering:

```
NPA
```

This command does all of the setup necessary to run NPA. The default attribute file NPAATTR is made local with this command along with the necessary library files needed for NPA command execution.

On NOS/VE only, you must make NPA available to your job with the following command:

```
CREATE_COMMAND_LIST_ENTRY $SYSTEM.CDCNET.VERSION_INDEPENDENT.COMMAND_LIBRARY
```

You then enter NPA and the following prompt appears:

```
np/
```

You may now enter NPA commands as well as system commands. To exit the utility (this disables NPA commands, and the np/ prompt no longer appears), enter:

```
quit
```

NOTE

If you are using NOS/VE, you can use the Concurrent Maintenance Library for the Virtual Environment (CML/VE) to interactively generate some of the NPA reports. See the CML/VE Reference manual or the CDCNET Hardware Installation and Troubleshooting manual for information on CML/VE usage.

If you are using NOS, you can use the Common Maintenance Software Interface (CMSI) to interactively generate some of the NPA reports. See the CML Reference manual or the CDCNET Hardware Installation and Troubleshooting manual for information on CMSI usage.

ARCHIVE_NPA_DATA_BASE (ARCNDB)

Purpose You use the ARCNDB procedure to remove records from the NPA databases and archive this information into an archive file.

Format

ARCHIVE_NPA_DATA_BASE
ARCHIVE_FILE=filename/reference
ARCHIVE_FILE_USER_NAME=username (NOS only)
DATA_BASE=keyword (or **list of keywords [NOS/VE only]**)
DATA_BASE_FILE_USER_NAME=username
BEGIN_DATE=yymmdd
BEGIN_TIME=hhmmss
END_DATE=yymmdd
END_TIME=hhmmss
OUTPUT=filename/reference
BEGIN_DATE_OFFSET=days
END_DATE_OFFSET=days
STATUS=status variable (NOS/VE only)

Parameters

ARCHIVE_FILE (AF)

Specifies the name of the file name (seven-character file for NOS or a standard file reference on NOS/VE) that receives the archived data.

NOTE

On NOS, this file must be a mass storage file. ARCNDB cannot archive directly to magnetic tape files.

ARCHIVE_FILE_USER_NAME (AFUN) (NOS only)

Specifies the user name (catalog) from which the archive file is to be acquired if the file is not already local to the current job. If the archive file is not found in the specified user catalog, a local file with the same name is created to hold the archived data. Default is the current user.

DATA_BASE (DB)

Specifies the NPA database you want to archive. Any one (or list for NOS/VE only) of the following keywords can be entered as a valid database value:

Keyword	Description
CONF	Configuration statistics database.
CONN	Connection statistics database.
DIOS	Device operating statistics database.
ETHR	Ethernet statistics database.
EVNT	Event log database.
HDLC	HDLC statistics database.
HERR	Hardware message database.

Keyword	Description
LOAD	Loader statistics database.
MCIS	Mainframe channel statistics database.
SERR	Software message database.
SESS	Session statistics database.
SUMM	Summary accounting statistics.
TELN	TELNET connection statistics database.
TERM	Terminal statistics database.
USER	User database.
X25C	X.25 connection statistics database.
ALL	All of the databases in this list.

Default value is ALL.

DATA_BASE_FILE_USER_NAME (DBFUN)

On NOS, this parameter specifies the user name from which the NPA databases are acquired. NPA databases must be permanent files to be archived. Local files of the same name as the databases are returned. Default is the current user name.

On NOS/VE, this parameter specifies the catalog from which the NPA databases are acquired. Databases must be permanent files to be archived. This file cannot be $LOCAL. Files of the same name as the databases in $LOCAL are detached. Default is the current working catalog, if permanent; otherwise $USER.

BEGIN_DATE (BD)

Specifies the earliest date from which records are to be archived (yy=year, mm=month, dd=day). Only those records recorded during or after the begin date are archived. Default is 000101.

BEGIN_TIME (BT)

Specifies the earliest time of day on the begin date from which the records are to be archived (hh=hour, mm=minute, ss=second). Only records recorded after the begin time on the begin date are archived. Default is 000000.

END_DATE (ED)

Specifies the last date from which records are to be archived (yy=year, mm=month, dd=day). Only records recorded on or before the end date are archived. Default is 991231.

END_TIME (ET)

Specifies the latest time of day on the end date that records are to be archived (hh=hour, mm=minute, ss=second). Only records recorded on or before the end time on the end date are archived. Default is 240000.

OUTPUT (O)

Specifies the output file (seven-character file name for NOS or standard file reference for NOS/VE) to receive diagnostic messages and processing statistics from the archive process. Default is ARCOUT for NOS and $LOCAL.ARCOUT for NOS/VE.

BEGIN_DATE_OFFSET (BDO)

Specifies the offset backwards in days (range of 0 through 365) from the current date. For example, a value of 1 indicates that the current day minus one, or yesterday, is the first day to begin archiving records. This parameter cannot be specified with the BEGIN_DATE parameter. Default is the BEGIN_DATE value.

END_DATE_OFFSET (EDO)

Specifies the offset backwards in days (range of 0 through 365) from the current date. For example, a value of 1 indicates that the current day minus one, or yesterday, is the last day that records are archived. This parameter cannot be specified with the END_DATE parameter. Default is the END_DATE value.

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples In this example, all data in the databases collected before 24:00 hours on February 8, 1985, are archived into the archive file, NPAARC.

```
ARCNDB,DB=ALL,AF=NPAARC,ED=850208
```

Aborts and Error Recovery

The Archive program takes the following action indicated for the condition specified:

1. Incorrect command syntax or values.
   - Issue appropriate error message to output file and dayfile.
   - Abort program.
2. Corrupt database file (NOS/VE only).
   - Write the diagnostic "xxxx DATA BASE FILE FOUND CORRUPTED" to the ARCNDB output file.
   - Recover all data up to the point of corruption either to the ARCHIVE_FILE or to the data retained in the database.
   - Write the diagnostic "xxxx RETAINED RECORDS RECOVERED TO NEW DATA BASE" to the ARCNDB output file. The resultant database file is not corrupt.

CHANGE_EXPECTED_OPERATING_LIMIT (CHAEOL)

Purpose

You use the CHAEOL command to change existing expected operating limits. On NOS, you must execute CHAEOL in full screen mode. The establishment of expected operations limits enables you to use NPA reports to detect unsatisfactory performance conditions in your network. Specific limits are defined for applicable reports. These limits are expressed as two numbers; an upper limit and a lower limit, both of which appear in the column heading portion of your report. Any reported values that do not fall within the specified limit range are called to your attention through the use of special symbols (< and >). If no data is available for a given time period, the character string "****" replaces the ending time field for that period.

CHAEOL initially displays the current limits for the report you specify. You may then change any or all of the limits for the report.

NOTE

On NOS, the CHAEOL command is only applicable in full screen mode.

Format

CHANGE_EXPECTED_OPERATING_LIMIT
REPORT_NAME=name
STATUS=status variable (NOS/VE only)

Parameters

REPORT_NAME (RN)

Specifies the name of the report for which you are changing expected operating limits. If there are no expected operating limits for the selected report, an error message is issued.

One of the following names must be entered as valid report name:

Name	Description
ETHRRP2	Ethernet report on frames and error statistics. Limits are expressed as numbers of instances.
MCISRP3	Mainframe channel report on bad/good block input/output statistics. Limits are expressed as percents.
TERMRP1	Terminal report on good/bad block input/output statistics. Limits are expressed as percents.

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples

In this example, the current NPA expected operating limits for report ETHRRP2 are displayed on your terminal screen. You change the current limits by entering new limits in their place.

```
CHAEOL,ETHRRP2
```

Aborts and Error Recovery

For NOS, the CCL interface responses to incorrect values are utilized for checking input parameters. The CCL procedure takes an abort exit if the application fails. This exit indicates that the application has failed. Responses from normal system job commands indicate the reason the job failed.

For NOS/VE, SCL sets the STATUS parameter to an abnormal condition.

CREATE_CDCNET_ANALYSIS_REPORT (CRECAR)

Purpose CRECAR is used to generate NPA reports. The parameters you use identify the reports you want to generate, the time interval that your reports cover, the identity of the system (DI) to which the reports apply, the output file name, whether the output file is added to or rewound, and the name of the catalog from which the database files are to be acquired.

Format

CREATE_CDCNET_ANALYSIS_REPORT
REPORT_NAME=report name (or **list of report names** [NOS/VE only])
DATA_BASE_FILE_USER_NAME=username
BEGIN_DATE=yymmdd
BEGIN_TIME=hhmm
END_DATE=yymmdd
END_TIME=hhmm
BEGIN_DATE_OFFSET=days
END_DATE_OFFSET=days
SYSTEM_ID=hexadecimal number (or *list of hexadecimal numbers* [NOS/VE only])
OUTPUT=filename/reference
APPEND=yes or *no* (NOS only)
LOG_ID=decimal number or *list of decimal numbers* [NOS/VE only])
EXCLUDE_LOG_ID=decimal number or *list of decimal numbers* [NOS/VE only])
SEVERITY=i, w, e, f, or *c*
COMPRESS=yes or *no*
STATUS=status variable (NOS/VE only)

Parameters

REPORT_NAME (RN)

On NOS, this parameter specifies the report name you want to generate. On NOS/VE, this parameter specifies a list of report names you want to generate. On both NOS and NOS/VE, the output is written to the file specified by the OUTPUT parameter.

One or more of the following keywords can be entered as valid report name values:

Keyword	Description
ALL	All reports are created except USERRP1.
DAILY	All daily reports are created. These include CONNRP2, DIOSRP2, DIOSRP4, HDLCRP2, MCISRP2, TELNRP2, and X25CRP2.
HOURLY	All hourly reports are created. These include CONNRP1, DIOSRP1, DIOSRP3, ETHRRP1, ETHRRP2, HDLCRP1, HDLCRP3, LOADRP1, MCISRP1, MCISRP3, SESSRP1, TELNRP1, TERMRP1, TERMRP2, and X25CRP1.

Keyword	Description
MESSAGS	All hardware and software message reports that are sorted by date and time are created. These include EVNTRP1, HRDWRP1, and SFTWRP1.
SEVERTY	All hardware and software message reports that are sorted by severity are created. These include EVNTRP2, HRDWRP2, and SFTWRP2.
STATIST	All statistical reports are created. These include CONNRP1, CONNRP2, DIOSRP1, DIOSRP2, DIOSRP3, DIOSRP4, ETHRRP1, ETHRRP2, HDLCRP1, HDLCRP2, HDLCRP3, LOADRP1, MCISRP1, MCISRP2, MCISRP3, SESSRP1, TELNRP1, TELNRP2, TERMRP1, TERMRP2, X25CRP1, and X25CRP2.
TERMINL	All terminal and connection reports are created. These include CONNRP1, CONNRP2, TERMRP1, and TERMRP2.
CONFRP1	Configuration report on an hourly status of DI hardware and software.
CONNRP1	Hourly connection report on the number of connections initiated and terminated.
CONNRP2	Daily connection report on the number of connections initiated and terminated.
DIOSRP1	Hourly device operating report on central processor unit (CPU) and memory utilization statistics.
DIOSRP2	Daily device operating report on CPU and memory utilization statistics.
DIOSRP3	Hourly device operating report on memory state transitions statistics.
DIOSRP4	Daily device operating report on memory state transitions statistics.
ETHRRP1	Hourly Ethernet report on transmit and collision channel statistics.
ETHRRP2	Hourly Ethernet report on frames and error statistics.
EVNTRP1	Event log report sorted by date and time.
EVNTRP2	Event log report sorted by severity.
EVNTRP3	Event log message frequency and severity summary.
EVNTRP4	Event log message frequency and severity summary reported by DI.

Keyword	Description
HDLCRP1	Hourly HDLC report on usage.
HDLCRP2	Daily HDLC report on usage.
HDLCRP3	Hourly HDLC report on error statistics.
HRDWRP1	Hardware message report sorted by date and time.
HRDWRP2	Hardware message report sorted by severity.
HRDWRP3	Hardware log message frequency and severity summary.
HRDWRP4	Hardware log message frequency and severity summary reported by a DI.
LOADRP1	Hourly loader statistics report.
MCISRP1	Hourly mainframe channel report on input/output block statistics.
MCISRP2	Daily mainframe channel report on input/output block statistics.
MCISRP3	Mainframe channel report on bad/good block input/output statistics.
SESSRP1	Hourly session statistics report on PDUs received and transmitted.
SFTWRP1	Software message report sorted by date and time.
SFTWRP2	Software message report sorted by severity.
SFTWRP3	Software log message frequency and severity summary.
SFTWRP4	Software log message frequency and severity summary reported by a DI.
TELNRP1	Hourly TELNET connection report for number of connections initiated, terminated, average connect time, and maximum connect time.
TELNRP2	Daily TELNET connection report for number of connections initiated, terminated, average connect time, and maximum connect time.
TERMRP1	Hourly terminal report on good/bad block input/output statistics.
TERMRP2	Hourly terminal statistics report online characters input/output.

Keyword	Description
USERRP1	Unsorted user message report.
X25CRP1	X.25 connection statistics report for number of connections initiated and terminated, average connect time, and maximum connect time.
X25CRP2	Daily X.25 connection statistics report for number of connections initiated, terminated, average connect time, and maximum connect time.

NOTE

No CRECAR report exists which draw upon data stored in the NPBSUMM database. See How to Create Customized NPA Reports Using IPF2 Database Files in the CDCNET Network Operations and Analysis manual for an example report derived from this database.

DATA_BASE_FILE_USER_NAME (DBFUN)

Specifies the user name (catalog) from which the database(s) are to be acquired. The databases must be permanent files (direct access file on NOS). If a local file exists with the same name as the database file you want, it is returned and then reattached. Default is the current user for NOS. For NOS/VE, default is the current working catalog if permanent; otherwise, $USER.

BEGIN_DATE (BD)

Specifies the date that begins your selected reporting period (yy=year, mm=month, dd=day). Default is 000101.

BEGIN_TIME (BT)

Specifies the time that report generation is to begin (hh=hour, mm=minute). This time applies only to records with the same date as the BEGIN_DATE parameter. Default is 0000.

END_DATE (ED)

Specifies the final date of the report period (yy=year, mm=month, dd=day). Default is 991231.

END_TIME (ET)

Specifies the time that report generation is to end (hh=hour, mm=minute). This applies only to records with the same date as the END_DATE parameter. Default is 2400.

BEGIN_DATE_OFFSET (BDO)

Specifies the offset backwards in days (range of 0 through 365) from the current date. For example, a value of 1 indicates that the current day minus one, or yesterday, is the first day to begin generating reports. This parameter cannot be specified with the BEGIN_DATE parameter. Default is the BEGIN_DATE value.

END_DATE_OFFSET (EDO)

Specifies the offset backwards in days (range of 0 through 365) from the current date. For example, a value of 1 indicates that the current day minus one, or yesterday, is the last day that reports are generated. This parameter cannot be specified with the END_DATE parameter. Default is the END_DATE value.

SYSTEM_ID (SID)

For NOS, this parameter specifies the last six hexadecimal digits of the system identification address of the DI for which the reports are being selected. The first six hexadecimal digits are always 080025. If the selected SID equals 000000, all SIDs are selected.

For NOS/VE, this parameter specifies a list of the last six hexadecimal digits of one to ten SIDs to include in the report. If one of the list of SIDs equals 000000, all SIDs are selected.

On both NOS and NOS/VE, if the parameter is selected, only log messages with SIDs matching the designated SIDs are used in creating reports. Default is all SIDs selected for inclusion in reports.

OUTPUT (O)

Specifies the file name (seven-character file name on NOS or standard file reference on NOS/VE) that is to contain the report. For NOS, the default value is CREOUT. For NOS/VE, the default value is $LOCAL.CREOUT.

NOTE

On NOS, this file is a Control Data display code file. It must be viewed interactively in normal mode or appropriately printed; otherwise, some data may appear garbled.

APPEND (NOS only)

YES specifies that the report created is to be directly written to the end of the output file. NO specifies that the output file is rewound before the created report is added. If the file is OUTPUT, reports are always appended. If not specified, the output file position remains as is.

LOG_ID (LI)

On NOS, this parameter specifies a single one- to five-digit log ID to select for report generation.

On NOS/VE, this parameter specifies a list of one to ten log IDs of one to five digits to select for report generation.

On both NOS and NOS/VE, this parameter is processed by the CONF, EVNT, HRDW, SFTW, and USER type reports only. When specified, only log messages with log IDs matching the designated log ID(s) are used in creating reports. When omitted, the selection of log IDs is determined by the presence or absence of the EXCLUDE_LOG_ID parameter. If both the LI and ELI parameters are omitted, all log IDs are selected for inclusion in the reports.

The LI and ELI may not be specified together.

EXCLUDE_LOG_ID (ELI)

On NOS, this parameter specifies a single one- to five-digit log ID to exclude from the reports.

On NOS/VE, this parameter specifies a list of one to ten log IDs of one to five digits to exclude from the reports.

On both NOS and NOS/VE, this parameter is processed by the CONF, EVNT, HRDW, SFTW, and USER type reports only. When specified, only log messages with log IDs matching the designated log ID(s) are excluded from the reports. When omitted, the selection of log IDs is determined by the presence or absence of the LOG_ID parameter. If both the LI and ELI parameters are omitted, all log IDs are selected for inclusion in the reports.

The LI and ELI may not be specified together.

SEVERITY (S)

Specifies the starting severity level to include in the reports. All log messages with this severity level or higher are reported. The highest level of severity is C (Catastrophic).

This parameter is processed by the EVNT, HRDW, and SFTW reports only. The default value is I (Informative). All severity levels are then included. Any of the following characters can be entered as a valid severity level:

Character	Severity
I	Informative
W	Warning
E	Error
F	Fatal
C	Catastrophic

COMPRESS (C)

YES specifies that reports are compressed. Default is NO, which specifies that no reports are compressed.

A compressed report does not contain a report heading page, data page headers, column headers or embedded page eject characters. It consists of an initial page eject followed by log message text (event type reports) or data (statistics reports).

This parameter increases report generation time.

STATUS (NOS/VE only)

This is the standard NOS/VE System Command Language (SCL) status variable. There is no default value.

Examples In this NOS example, report ETHRRP2 is created to report statistics generated from the DI with the system ID of 123456 between 00:00 hour on January 1, 1985, and 24:00 January 2, 1985. ETHRRP2 is added to the end of the default output file CREOUT. The database file user name is the current user.

```
CRECAR,RN=ETHRRP2,BD=850101,ED=850102,SID=123456,APPEND=yes
```

Aborts and Error Recovery

For NOS, the CYBER Control Language (CCL) interface responses to incorrect values are utilized for checking input parameters. The CCL procedure takes an abort exit if the application fails. This exit indicates that the application has failed. Responses from normal system job commands indicate the reason the job failed.

For NOS/VE, the SCL sets the STATUS parameter to an abnormal condition.

EDIT_CDCNET_LOG_MESSAGE (EDICLM)

Purpose You use EDICLM to create, add to, or change the site information section of a CDCNET log message.

Format

EDIT_CDCNET_LOG_MESSAGE
MESSAGE_NUMBER=log id number
EDITOR=type (NOS only)
STATUS=status variable (NOS/VE only)

Parameters

MESSAGE_NUMBER (MN)

Specifies the log identifier number of the log message to be edited. On NOS, this parameter must consist of five digits and must be zero-filled on the left (a log message number of 984 is represented as 00984.)

EDITOR (E) (NOS only)

Specifies the type of editor to be used during your edit. Valid types are FSE, EDIT, or XEDIT. Default is FSE.

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples In this NOS example, log message number 00001 appears on your terminal screen, and you may edit the site information portion of the message using FSE.

```
EDICLM,MN=00001,FSE
```

EXPLAIN_CDCNET_LOG_MESSAGE (EXPCLM)

Purpose You use EXPCLM to receive the following information on CDCNET log messages:

- The purpose of the message
- Actions required, if any
- Site information concerning the log message

The message purpose section is an expanded explanation of your CDCNET log message. The actions required section tells you the action to be taken for the log message. The site information is provided by your site and contains any information you consider relevant to the log message.

Format

EXPLAIN_CDCNET_LOG_MESSAGE
MESSAGE_NUMBER=log id number or **LIST**
STATUS=status variable (NOS/VE only)

Parameters

MESSAGE_NUMBER (MN)

Specifies the log message number of the log message for which you want information. For NOS, this parameter must consist of five digits and must be zero-filled on the left. (A log message number of 984 is represented as 00984.) If LIST is selected, it specifies that all the log message numbers with their associated names are listed.

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples In this example, log message number 00001 appears on your terminal screen.

```
EXPCLM,MN=00001
```

Aborts and Error Recovery

For NOS, the CCL interface responses to incorrect values are utilized for checking input parameters. The CCL procedure takes an abort exit if the application fails. This exit indicates that the application has failed. Responses from normal system job commands indicate the reason the job failed.

For NOS/VE, the SCL sets the STATUS parameter to an abnormal condition.

REFORMAT_CDCNET_LOG_FILE (REFCLF)

Purpose REFCLF is used to execute the reformatting process. REFCLF performs the following:

- Converts network log file records into database records
- Transfers the converted records into appropriate data files

REFCLF generates the following databases:

NPA Database	Definition	Type of Database
NPBACNT	Accounting information	Account
NPBCONF	Configuration messages	Event
NPBCONN	Connection statistics	Event
NPBDIOS	Device operating statistics	Statistic
NPBETHR	Ethernet statistics	Statistic
NPBEVNT	Event log messages	Event
NPBHDLC	High-Level Data Link Channel (HDLC) statistics	Statistic
NPBHERR	Hardware messages	Event
NPBLOAD	Loader statistics	Event
NPBMCIS	Mainframe channel statistics	Statistic
NPBSERR	Software messages	Event
NPBSESS	Session statistics	Statistic
NPBSUMM	Summary statistics	Statistic
NPBTELN	TELNET connection statistics	Statistic
NPBTERM	Terminal statistics	Statistic
NPBUSER	User messages	Event
NPBX25C	X.25 connection statistics	Event

NOTE

REFCLF does not support full screen execution.

Format **REFORMAT_CDCNET_LOG_FILE** (A period must follow the command for NOS. A comma or blank must follow the command for NOS/VE.)

LOG_FILE=filename/reference or *list of filename/reference*
LOG_FILE_USER_NAME=name (NOS only)
ATTRIBUTE_FILE=filename/reference
ATTRIBUTE_FILE_USER_NAME=name (NOS only)
DATA_BASE=value or *list of value*
DATA_BASE_FILE_USER_NAME=name
USER=attribute(s) or *ALL*
BEGIN_DATE=yymmdd
BEGIN_TIME=hhmmss
END_DATE=yymmdd
END_TIME=hhmmss
STATISTICS_INTERVAL=value
OUTPUT=filename/reference
LOG_FILE_LIST=filename/reference
STATUS=status variable (NOS/VE only)

Parameters *LOG_FILE (LF)*

Specifies a filename (seven-character file name on NOS or standard file reference on NOS/VE) or list of CDCNET log files (filenames) that REFCLF processes. A maximum of 10 log files can be specified. On NOS, if no log file is specified, the reformatter finds up to 30 files beginning with NL or NETLFmid and uses them. On NOS/VE, if no log file is specified, the reformatter accesses and processes up to 30 cycles of the $SYSTEM.CDCNET.LOG file.

LOG_FILE_USER_NAME (LFUN) (NOS only)

Specifies the user name (catalog) from which the log files are to be acquired. If the file is already local to the job, no acquire is performed. Default is SYSTEMX.

ATTRIBUTE_FILE (AF)

Specifies the attribute file (seven-character file name on NOS or standard file reference on NOS/VE) that contains attribute information for the log messages. The attribute information tells REFCLF where to collect specific log entries. If no attribute file is specified, the standard NPAATTR file is used. The access command NPA makes the standard NPAATTR file local to the job.

The attribute file is a text file of log message identifier (ID) numbers and associated attributes in the following format:

```
nnn aa aa ...
```

nnn is the log message ID number in decimal format, and aa is a one- to four-character attribute code associated with the log message (blanks are used as separators between fields). A log message can have one or more associated attributes. The following are valid attributes:

Attribute	Description
A	Accounting
HE	Hardware messages
SE	Software messages
S	Statistics
EL	Events log
x..xxxx	User-defined one to four alphanumeric characters (cannot be ALL)

NOTE

REFCLF accepts both sorted and unsorted attribute files. However, REFCLF processing time increases if the file is unsorted. Therefore, presenting REFCLF with a sorted file is recommended.

The attributes provide the association between a log message and an NPA database. A log message is reformatted to a particular database or databases, depending on its attributes, as follows:

Attribute	Database
A	NPBACNT
HE	NPBHERR
SE	NPBSERR
S	(See note)
EL	NPBEVNT
x..xxxx	NPBUSER

NOTE

The S (statistics) attribute is valid only for specific predefined log messages. These messages are predefined to NPA and targeted to certain related databases. For example, log message 166 is a terminal statistics message that, with an attribute of S, is directed to NPBTERM. An S attribute for log messages other than those that are predefined has no effect. The log message is discarded if no other attribute is specified.

Log messages with user-defined attributes (X..XXXX) may only be selected by use of the REFCLF USER (U) parameter. If a log message has an identifier which is not in the attribute file, the message is discarded unless the USER (U) parameter is set to ALL. Setting USER to ALL directs all log messages to the NPBUSER database.

ATTRIBUTE_FILE_USER_NAME (AFUN) (NOS only)

Specifies the user name (catalog) from which the directive file is acquired. If this file is already local to the job, no acquire is performed. The access command NPA makes the standard NPAATTR file local to the job. Default is current user.

DATA_BASE (DB)

Enter one or more values from the following value list. Either the DB parameter or the USER parameter must be specified. If only the USER parameter is specified, the default is none.

Value	Description
ALL	All of the messages and statistics in this list except ACNT are written to their respective files.
ACNT	Accounting log messages are written to the NPBACNT file. If this option is not specified, the accounting file is not written.
CONF	Device interface (DI) configuration log messages are written to the NPBCONF file. If this option is not specified, the DI configuration file is not written.
CONN	DI connection log messages are written to the NPBCONN file. If this option is not specified, the DI connection file is not written.
DIOS	DI operating statistics log messages are written to NPBDIOS file. If this option is not specified, the DI operating statistics file is not written.
ETHR	Ethernet statistics information is written to the NPBETHR file. If this option is not specified, the Ethernet statistics are not written.
EVNT	DI event log messages are written to the NPBEVNT file. If this option is not specified, the DI event log messages are not written.
HDLC	HDLC statistic log messages are written to the NPBHDLC file. If this option is not specified, the HDLC statistics file is not written.
HERR	DI hardware errors are written to the NPBHERR file. If HERR is not specified, the DI hardware errors are not written.
LOAD	System loader statistics information is written to the NPBLOAD file. If this option is not specified, the system loader statistics are not written.
MCIS	Mainframe channel statistics are written to the NPBMCIS file. If MCIS is not specified, the mainframe channel statistics are not written.
SERR	DI software errors are written to the NPBSERR file. If SERR is not specified, the DI software errors are not written.

Value	Description
SESS	The session layer statistic log message is written to the NPBSESS file. If this option is not specified, the session layer statistics file is not written.
SUMM	Connection statistics for Terminal Support, X.25 AsyncTip, Server TELNET TIP, X.25 Terminal Gateway, X.25 Gateway, Passthrough, and Device Outcall are written to the file NPBSUMM. If SUMM is not specified, the NPBSUMM file is not written.
TELN	TELNET connection log messages are written to the NPBTELN file. If TELN is not specified, the TELNET connection file is not written.
TERM	Terminal statistics are written to the NPBTERM file. If not specified, the terminal statistics are not written.
X25C	X.25 connections are written to the NPBX25C file. If X25C is not specified, the NPBX25C file is not written.
ALL	All of the messages and statistics in this list except ACNT are written to their respective files.

DATA_BASE_FILE_USER_NAME (DBFUN)

On NOS, this parameter specifies the user name from which the NPA database files are to be acquired. If a database file is already local to the job, then no acquire is performed. If a database file is not found, it is created as a local file. Default is the current user name.

On NOS/VE, this parameter specifies the catalog from which the NPA database files are to be acquired. If a database file is not found, it is created in the specified (or default) catalog. Default is the current working catalog, if permanent; otherwise, $USER.

USER (U)

Specifies that the NPBUSER database is to be written. The purpose of this parameter is for diagnosis of log file contents. The attribute, or attributes, specified determine which log messages are to be selected. Values consist of one or more attributes listed in the attribute file or ALL. If the value ALL is specified, all log messages are written to the USER database. Either the USER or DB parameter must be specified. If only the DB parameter is specified, the USER database is not written.

BEGIN_DATE (BD)

Specifies the beginning date of the log messages to be reformatted (yy=year, mm=month, dd=day). Only messages with a date that is equal to or greater than this parameter are reformatted. Default is 000101.

BEGIN_TIME (BT)

Specifies the beginning time of the log messages to be reformatted (hh=hours, mm=minutes, ss=seconds). Only messages with a time equal to or greater than begin date are reformatted. Default is 000000.

END_DATE (ED)

Specifies the end date of the log messages to be reformatted (yy=year, mm=month, dd=day). Only messages with a date equal to or less than this parameter are reformatted. Default is 991231.

END_TIME (ET)

Specifies the ending time of the log messages to be reformatted (hh=hour, mm=minutes, ss=seconds). For log messages with a date equal to the ending date, only messages with a time less than or equal to this value are reformatted. Default is 240000.

STATISTICS_INTERVAL (SI)

For this release and future releases, this parameter is ignored.

OUTPUT (O)

Specifies the output file (seven-character file name on NOS or standard file reference on NOS/VE) to which REFCLF writes diagnostic and processing statistics. Default is REFOUT for NOS. Default is $LOCAL.REFOUT for NOS/VE.

LOG_FILE_LIST (LFL)

Specifies a file name (seven-character file name on NOS or standard file reference on NOS/VE) to which a list of completely processed log file filenames is written. The same log file list plus an informative message are also written to the REFCLF Output (O) file. The LOG_FILE_LIST file is in a format suitable for site-written procedures to know the log files successfully processed by REFCLF. Site procedures can use this information to dump and purge the successfully processed log files. Default is no file written. The LFL file format is as follows:

```
file name/reference,
  :
file name/reference
```

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples

1. This NOS example shows reformatting of all log files (up to 30) in the SYSTEMX user catalog. This includes the CDCNET NETLFmid log file and all log files with names beginning with NL. All log messages with attributes matching those defined in the NPAATTR default attribute file are reformatted. All databases in the user catalog, except NPBACNT and NPBUSER, are updated if corresponding log messages are encountered. Information pertaining to this REFCLF run is written to the REFOUT default file.

```
REFCLF.DB=ALL
```

2. In this NOS example, REFCLF is used to reformat log messages in the three log files NL09AA, NL09BB, and NL09CC in the SYSTEMX user catalog. Only log messages with attributes matching those in the NPAATTR default attribute file and log messages falling in the interval from noon September 3, 1985, until midnight September 7, 1985, are processed. Only databases NPBACNT, NPBCONN, NPBETHR, and NPBSERR in the user catalog are updated. The results of the REFCLF run are written to the REFREP file, and a list of completely processed log files (if any) are written to the PURGLST file.

```
REFCLF.LF=(NL09AA,NL09BB,NL09CC),BD=850903,BT=120000,ED=
850907,DB=(ACNT,CONN,ETHR,SERR),O=REFREP,LFL=PURGLST
```

3. This NOS example shows REFCLF being used to extract four log messages from the file NLB0910 in catalog LOGCAT and write them to the NPBACNT database in catalog DBCAT. Local file ACNTATR is a copy of the NPAATTR default attribute file, which has been edited to only contain entries for the four log messages. The A accounting attribute has been added to each one in the ACNTATR file, as shown in the following list:

```
00010    A    HE    EL
00017    A    EL
00424    A    S
00427    A    SE    EL
```

NOTE

Only the NPBACNT database is updated or created when this example is executed.

```
REFCLF.LF=NLB0910,LFUN=LOGCAT,DB=ACNT,DBFUN=DBCAT,AF=ACNTATR
```

4. This example (figure 18-1) illustrates the command parameters that appear in the output file from a REFCLF data reformat run on NOS.

```
Network Performance Analyzer Version xxxx 1989/11/11 1:57 PM Page -   1

** Network Performance Analyzer -- Reformat Command Parameters **

Attribute file --> NPAATTR

Attribute file user name --> NETADMN

Log file list --> FILELST

Output file --> REFOUT

Begin Date --> 000101
Begin Time --> 000000
End Date   --> 991231
End Time   --> 240000

Log file (   1 ) --> LOG1R11
Log file (   2 ) --> NLA0129

Log file(s) user name --> SYSTEMX

Data base file(s) user name --> NETADMN

Data Base file --> acnt
Data Base file --> conf
Data Base file --> conn
Data Base file --> dios
Data Base file --> ethr
Data Base file --> evnt
Data Base file --> exts
Data Base file --> hdlc
Data Base file --> herr
Data Base file --> load
Data Base file --> mcis
Data Base file --> queu
Data Base file --> serr
Data Base file --> sess
Data Base file --> summ
Data Base file --> teln
Data Base file --> term
Data Base file --> user
Data Base file --> x25c
```

Figure 18-1. Output File Command Parameters

5. This example (figure 18-2) illustrates the attribute file that appears in the output file after a REFCLF run.

```
Network Performance Analyzer Version xxxx 1989/11/11 1:57 PM Page -   2

** Network Performance Analyzer -- Attribute File **

00018 EL
00019 EL
00020 EL
00021 EL
00022 EL
00057 EL
00058 EL SE
00059 EL SE
00060 EL SE
00061 EL SE
00062 EL SE
00063 EL SE
00064 EL SE
00065 EL SE
00066 EL SE
00067 EL
00068 EL
00069 EL SE
00070 EL SE
00071 EL SE
00072 EL SE
00073 EL SE
00089 EL
00090 EL
00091 SE
00092 SE
00093 SE
00094 S
00161 SE EL
00162 SE EL
00163 SE EL
00164 EL
00165 EL


**End of Attribute File **
```

Figure 18-2. Output File Attribute File

6. This example (figure 18-3) illustrates the error messages that may appear in the output file after a REFCLF run.

```
1. The following messages report difficulties in processing log file protocol
   data units.

   PROBLEM ENCOUNTERED IS: type of problem encountered
   EXPECTING THE FOLLOWING FIELD: field name expected
   CURRENT LOG FILE PDU NUMBER PROCESSING: xxxxxx
   CURRENT LOG ID: xxxx  DATE/TIME: yy/mm/dd  hh.mm.sssss
   LOG FILE IS: xxxxxxx TOTAL PDU LENGTH: xxxxxx  PDU HEADER LENGTH: xxxxxx
     PDU BODY LENGTH: xxxxxx CURRENT BYTE IN PDU BODY: xxxxxx
   RECOVERY ATTEMPTED: type of recovery attempted

   INFORMATIVE MESSAGE: LOG MESSAGE nnnnn REFORMATTING INCOMPLETE.
                        MDU PARAMETER STRING > 256 CHARACTERS.
                        DATE/TIME: yy/mm/dd  hh.mm.sssss

2. On NOS, the following error messages are written to the output file if
   REFCLF encounters errors while attempting to access the database files.

   DATA BASE NOT FOUND OR NO WRITE PERMIT. LOCAL FILE CREATED --> LFN = filename

   DATA BASE ACCESS ERROR, PROCESSING BYPASSED --> LFN = filename

   DATA BASE BUSY, PROCESSING BYPASSED --> PFN = filename, UN = username

3. The following message occurs when an attempt is made to process an MDU
   whose integer value is greater than 48 bits on NOS or 64 bits on NOS/VE.

   INFORMATIVE MESSAGE: LOG MESSAGE xxxx CONTAINS AN MDU WHOSE
                        INTEGER REPRESENTATION IS > nn BITS.
                        AN INTEGER VALUE OF 0 HAS BEEN WRITTEN
                        TO THE DATABASE.
                        DATE/TIME: yy/mm/dd hh.mm.sssss

nn = 48 (NOS) or 64 (NOS/VE)
```

Figure 18-3. Output File Error Messages

(Continued)

(Continued)

```
4. The following error messages are encountered while processing the
   attribute file.

   ERROR - NO ATTRIBUTE FILE FOUND.

   ERROR - ATTRIBUTE FILE IS EMPTY.

   ERROR - THE FOLLOWING ATTRIBUTE FILE LINE HAS A LOG ID OUT OF RANGE

   ERROR - THE FOLLOWING ATTRIBUTE FILE LINES ARE IGNORED

   ERROR - TOO MANY LOG IDS HAVE BEEN DEFINED.  CURRENT MAX IS 1228

   ERROR - THE ATTRIBUTE FILE CONTAINS THE FOLLOWING DUPLICATE LOG IDS -
     xxxxx  xxxxx . . .

   INFORMATIVE - THE FOLLOWING ATTRIBUTE FILE LINE HAS AN ATTRIBUTE CODE
                 WHICH IS GREATER THAN 4 CHARACTERS IN LENGTH.  FIRST 4
                 CHARACTERS ARE USED.

   INFORMATIVE - THE FOLLOWING ATTRIBUTE FILE LINE HAS A LOG ID WITH NO
                 ATTRIBUTES

   ** BAD ATTRIBUTE FILE - REFCLF ABORT **
```

Figure 18-3. Output File Error Messages

7. This example (figure 18-4) illustrates the text that appears in the REFCLF output file if the LFL parameter is specified. For this example, the LFL file (FILELST) contains the following text:

```
LOG1R11,
NLA0129
```

```
Network Performance Analyzer Version 1.10/4103  1989/11/11 1:57 PM Page - xx

The following log files have been completely processed to the CONF,
CONN, DIOS, ETHR, EVNT, EXTS, HERR, HDLC, LOAD, MCIS, QUEU, SERR, SESS,
SUMM, TELN, TERM, and X25C databases:

  LOG1R11,
  NLA0129
```

Figure 18-4. Output File Processed Log Files

8. The following two examples (figure 18-5 and 18-6) show the information that appears in the output file for statistics pertaining to the REFCLF reformatting run.

   Figure 18-5 shows the number of log entries selected for reprocessing to each individual database and the number of resultant database records written. A single selected log entry may result in multiple database records being written to a particular database. A single log entry may also be selected for reformatting to multiple databases. For these reasons, the number of database records written does not necessarily match the number of log entries selected; and the total number of log entries selected does not necessarily match the total log entries processed for the individual log files.

   Figure 18-6 is a summary, by log file, of the number of log entries read, the number of log entries processed, and the number of log entries rejected. The time and date of the earliest and latest processed log entries on the log file are also listed. The number of log entries read represents the total number of log entries on the log file. The number of log entries processed represents the total number of log entries whose attributes and time/date stamps match the specified REFCLF command parameter values (database selected, begin, and end time/date) and which have been successfully reformatted and written to one or more databases. The number of log entries rejected is represented by four totals: ERRORS ENCOUNTERED, NOT IN ATTRIBUTE FILE, NOT SELECTED -> DATABASE, and OUTSIDE TIME/DATE WINDOW. ERRORS ENCOUNTERED reflects those log entries which could not be reformatted due to errors. NOT IN ATTRIBUTE FILE reflects those log entries which were not found in the attribute file. NOT SELECTED -> DATABASE reflects those entries which were not reformatted because they are not targeted to databases specified in the REFCLF command. OUTSIDE TIME/DATE WINDOW reflects those log entries whose time/date stamps fall outside the user specified time/date window. A rejected log entry is included in only one of the four totals.

   An additional total may appear if reformatting to the NPBSUMM database has been requested and both old (pre L750) and new versions of connection statistics log messages 38, 39, 617-620, 1160, 1161, and 1340-1343 are read from the log file(s). This total represents the number of old messages read. The REFCLF NPBSUMM reformatter ignores old versions of these log messages because they contain incomplete data.

   Performance information follows the log file summaries. CPU time used, elapsed time used and log entries read, processed and selected and database records written per unit time are listed. Memory allocation information is also listed.

```
Network Performance Analyzer Version 1.10/4103  1989/11/11  1:57 PM Page -  xxx

** Network Performance Analyzer -- Statistics **

            DATABASE  LOG
 DATABASE   RECORDS   ENTRIES
 NAME       WRITTEN   SELECTED
 -------    -------   --------
 NPBACNT          0          0
 NPBCONF        833        308
 NPBCONN       7187       7187
 NPBDIOS        721        721
 NPBETHR        379        379
 NPBEVNT      10605      10605
 NPBEXTS          0          0
 NPBHERR         58         58
 NPBHDLC         68         68
 NPBLOAD        107        107
 NPBMCIS        320        320
 NPBQUEU          0          0
 NPBSERR        948        948
 NPBSESS        341        341
 NPBSUMM      12568      12568
 NPBTELN          0          0
 NPBTERM        508        508
 NPBUSER          0          0
 NPBX25C       2962       2962
             -------    -------
 TOTAL        37605      37080

 LOG FILE PROCESSING:

    LOG FILE --> LOG1R11

        FILE STATUS: FILE FOUND IN SPECIFIED CATALOG.
        DATE/TIME OF FIRST PROCESSED ENTRY: 86/01/01  00.02.49049
        DATE/TIME OF LAST  PROCESSED ENTRY: 86/09/06  03.20.26026
        LOG ENTRIES READ:                5723
        LOG ENTRIES PROCESSED:           5717
        LOG ENTRIES REJECTED:
          ERRORS ENCOUNTERED:               0
          NOT IN ATTRIBUTE FILE:            6
          NOT SELECTED -> DATABASE:         0
          OUTSIDE TIME/DATE WINDOW:         0
```

Figure 18-5. Output File Log Entries

```
Network Performance Analyzer Version 1.10/4103  1989/11/11  1:57 PM Page -  xxx

    LOG FILE --> NLA0129

        FILE STATUS: FILE FOUND IN SPECIFIED CATALOG.
        DATE/TIME OF FIRST PROCESSED ENTRY: 87/01/28  03.14.50189
        DATE/TIME OF LAST  PROCESSED ENTRY: 87/01/29  12.46.53717
        LOG ENTRIES READ:            49055
        LOG ENTRIES PROCESSED:       16301
        LOG ENTRIES REJECTED:
          ERRORS ENCOUNTERED:            0
          NOT IN ATTRIBUTE FILE:        24
          NOT SELECTED -> DATABASE:  32730
          OUTSIDE TIME/DATE WINDOW:      0

 TOTALS: LOG ENTRIES READ:           54778
         LOG ENTRIES PROCESSED:      22018
         LOG ENTRIES REJECTED:  32760

REFCLF CPU PROCESS TIME              =    473.652 CPU SECONDS

LOG ENTRIES READ PER CPU SECOND      =    115 ENTRIES/CPU SECOND

LOG ENTRIES PROCESSED PER CPU SECOND =    115 ENTRIES/CPU SECOND

LOG ENTRIES SELECTED PER CPU SECOND  =     49 ENTRIES/CPU SECOND

NPA RECORDS WRITTEN PER CPU SECOND   =     52 ENTRIES/CPU SECOND

REFCLF ELAPSED TIME                  =    819 ELAPSED SECONDS

LOG ENTRIES READ PER SECOND          =     66 ENTRIES/CPU SECOND

LOG ENTRIES PROCESSED PER SECOND     =     66 ENTRIES/CPU SECOND

LOG ENTRIES SELECTED PER SECOND      =     28 ENTRIES/CPU SECOND

NPA RECORDS WRITTEN PER SECOND       =     30 ENTRIES/CPU SECOND

NUMBER OF PDU PUFFER ALLOCATIONS     =      4

AVERAGE BUFFER SIZE =  4154

MAXIMUM BUFFER SIZE =  6616

REFCLF PROCESSING COMPLETE
```

Figure 18-6. Output File Statistics Summary

9. This example (figure 18-7) illustrates the dayfile messages that may appear in the output file after a REFCLF run. In general, REFCLF actions and error messages are left-justified. REFCLF tasks to be completed are indented. The terms ACCOUNTING, STATISTICS, EVENT, and USER correspond to the basic NPA record types.

```
1.  Messages produced as a result of REFCLF command errors:

    ERROR, PARAMETERS DB OR USER MUST BE SPECIFIED
    ERROR, REFCLF MUST BE FOLLOWED BY A PERIOD
    FORMAT ERROR ON COMMAND
    ILLEGAL LOG FILE NAME xxxxxxx
    INCORRECT REFCLF CONTROL CARD
    INVALID DATE: reason
    INVALID TIME: reason
    NO LOG FILES TO PROCESS
    TOO MANY LOG FILES. > nn
    UNKNOWN PARAMETER xxxxxxx

2.  Messages produced during the reformatting process:

    BEGIN REFCLF PROCESSING
       COPY ACCOUNTING LOG MESSAGES
       REFORMAT STATISTICS LOG MESSAGES
       REFORMAT EVENT LOG MESSAGES
       REFORMAT USER LOG MESSAGES
    ERROR - ATTRIBUTE FILE HAS LOG ID OUT OF RANGE
    ERROR - THE ATTRIBUTE FILE CONTAINS DUPLICATE LOG IDS
    ERROR - NO ATTRIBUTE FILE FOUND
    ERROR - ATTRIBUTE FILE EMPTY
    ERROR - TOO MANY LOG IDS HAVE BEEN DEFINED
    INFORMATIVE - ATTRIBUTE FILE HAS LOG ID WITH NO ATTRIBUTES
    INFORMATIVE - ATTRIBUTE FILE HAS CODE > 4 CHARACTERS
    SORTING THE ATTRIBUTE FILE CONTENTS
    DATA BASE ACCESS ERROR, PROCESSING BYPASSED --> LFN = filename
    DATA BASE BUSY, PROCESSING BYPASSED --> PFN = filename, UN = username
    PROCESSING LOG FILE: log file
    LOG FILE READ ERROR-PROCESSING STOPPED: log file
    LOG FILE EMPTY OR NOT FOUND: log file
    LOG FILE READ ERROR-PROCESSING STOPPED
       COMPLETE NPA DATABASE GENERATION
       PREPARE REFCLF STATISTICS
    REFCLF PROCESSING COMPLETE
```

Figure 18-7. Dayfile Messages

Aborts and Error Recovery

REFCLF takes the following action for the condition specified:

1. Incorrect command syntax or parameter values.
   - Issue appropriate error message to output file and dayfile.
   - Abort program.
2. Parameter specified more than once in REFCLF command.
   - Issue informative message.
   - Use last value specified (NOS only).
   - Give user a choice of values to use (NOS/VE only).
3. End-of-file is encountered unexpectedly (NOS only).
   - Issue appropriate error message to output file and dayfile.
   - Terminate processing the current log file. Process other log files, if any.
4. End-of-record is encountered unexpectedly (NOS only).
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
5. Unexpected management data unit (MDU) types (for example, an integer value is expected in the log protocol data unit [PDU], but a string value was encountered).
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
6. The log PDU is not in correct format (that is, REFCLF does not recognize a field in the log PDU).
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
   - Skip record forward is performed on the current log file (NOS only).
7. The log header is not in correct format (NOS only).
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
   - Skip record forward is performed on the current log file.
8. There is not enough memory to hold a log PDU (NOS only).
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
   - Skip record forward is performed on current log file.

9. Log PDU contains less than the expected number of MDUs.
   - Issue appropriate error message to output file.
   - The log PDU is ignored.
10. Attribute file is empty or not found.
   - Issue appropriate informative error message to output file.
   - Abort program.
11. The log identifier specified in the attribute file is out-of-range, too many log identifiers have been specified, or duplicate log identifiers exist.
   - Issue appropriate warning error message to output file.
   - Abort program.
12. Too many log files. > 10.
   - Issue appropriate warning error message to output file.
   - Abort program.

   NOTE

   Too many log files occurs when more than 10 log files are specified on the command; or on NOS, no log file was specified but more than 30 log files were found in user name LFUN (SYSTEMX if not specified); or on NOS/VE, no log file was specified but more than 30 log file cycles were found for $SYSTEM.CDCNET.LOG.

   When this occurs, make separate REFCLF runs specifying 10 or fewer log files.

13. No log files to process. No log files were specified and REFCLF could not find any to process after searching user name LFUN (NOS) or $SYSTEM.CDCNET (NOS/VE).
   - Issue appropriate warning error message to output file.
   - Abort program.

RELOAD_NPA_DATA_BASE (RELNDB)

Purpose You use the RELNDB procedure to reload records from an archive file and merge these files into existing NPA databases.

Format

RELOAD_NPA_DATA_BASE
ARCHIVE_FILE=filename/reference
ARCHIVE_FILE_USER_NAME=username (NOS only)
DATA_BASE=keyword (or list of keywords [NOS/VE only])
DATA_BASE_FILE_USER_NAME=value
BEGIN_DATE=yymmdd
BEGIN_TIME=hhmmss
END_DATE=yymmdd
END_TIME=hhmmss
OUTPUT=filename/reference
STATUS=status variable (NOS/VE only)

Parameters

ARCHIVE_FILE (AF)

Specifies the name of the file (seven-character file name for NOS or a standard file reference name for NOS/VE) from which the archived data is retrieved.

ARCHIVE_FILE_USER_NAME (AFUN) (NOS only)

Specifies the user name (catalog) from which the archive file is to be acquired, if not already local to the job. Default is the current user.

DATA_BASE (DB)

Specifies the database (keyword) you want to reload. Any one (or list for NOS/VE only) of the following keywords can be entered as a valid database value:

Keyword	Description
CONF	Configuration statistics database.
CONN	Connection statistics database.
DIOS	Device operating statistics database.
ETHR	Ethernet statistics database.
EVNT	Event log database.
HDLC	HDLC statistics database.
HERR	Hardware message database.
LOAD	Loader statistics database.
MCIS	Mainframe channel statistics database.
SERR	Software message database.
SESS	Session statistics database identifier.
SUMM	Summary statistics database.

Keyword	Description
TELN	TELNET connection statistics database.
TERM	Terminal statistics database.
USER	User database.
X25C	X.25 connection statistics database.
ALL	All of the databases in this list.

Default is ALL.

DATA_BASE_FILE_USER_NAME (DBFUN)

On NOS, this parameter specifies the user name from which the NPA databases are acquired. NPA databases must be permanent files to be reloaded. Local files of the same name as the databases are returned. Default is the current user.

On NOS/VE, this parameter specifies the catalog from which the NPA databases are to be acquired. If a database is not found, it is created in the specified (or default) catalog. The NPA databases must be permanent files to be reloaded. This parameter cannot be $LOCAL. Files of the same name as the database in $LOCAL are detached. Default is $CATALOG if permanent; otherwise $USER.

BEGIN_DATE (BD)

Specifies the earliest date (yy=year, mm=month, dd=day) from which records are to be reloaded. Only those records archived during or after the begin date are reloaded. Default is 000101.

BEGIN_TIME (BT)

Specifies the earliest time of day (hh=hour, mm=minute, ss=second) on the begin date from which the records are to be reloaded. Only records recorded after the begin time on the begin date are reloaded. If no begin time is entered, records recorded after 00:00 hour on the begin date are reloaded.

END_DATE (ED)

Specifies the last date (yy=year, mm=month, dd=day) from which records are to be reloaded. Only records recorded on or before the end date are reloaded. Default is 991231.

END_TIME (ET)

Specifies the latest time of day (hh=hour, mm=minute, ss=second) on the end date that records are to be reloaded. Only records recorded on or before the end time on the end date are reloaded. If no end time is entered, records recorded before 24:00 hours on the end date are reloaded.

OUTPUT (O)

Specifies the output file (seven-character file name for NOS or a standard file reference name for NOS/VE) to receive diagnostic messages and processing statistics from the reload process. Default is RELOUT for NOS. Default is $LOCAL.RELOUT for NOS/VE.

STATUS (NOS/VE only)

This is the standard NOS/VE SCL status variable. There is no default value.

Examples In this example, all of the database information written to the archive file NPAARC between 00:00 hour on January 8, 1985, and 24:00 on February 8, 1985, is reloaded into the databases.

```
RELNDB,DB=ALL,AF=NPAARC,BD=850108,ED=850208
```

Aborts and Error Recovery

The Reload program takes the following action indicated for the condition specified:

1. Incorrect command syntax or values.
   - Issue appropriate error message to output file and dayfile.
   - Abort program.
2. Corrupt database file (NOS/VE only).
   - Write the diagnostic "xxxx DATA BASE FILE FOUND CORRUPTED" to the RELNDB output file.
   - Recover all data up to the point of corruption to the database.
   - Write the diagnostic "NUMBER OF RECORDS RECOVERED--> nnnn" to the RELNDB output file. The resultant database file is not corrupt.

SUBMIT BATCH JOB PERIODICALLY (SUBBJP) NOS MDI Only

Purpose

You execute SUBBJP in batch mode to regulate the periodic execution of NPA commands.

SUBBJP allows you to submit job files to the input queue for processing at a periodic rate. The input parameter you specify identifies a batch directive file. The batch directive file contains one or more records each of which defines:

- Name of a job file to submit to the input queue
- User name of the job file to submit to the input queue
- Start date and time to begin the periodic interval
- Periodic interval, specified in days, hours, and minutes
- Last date and time this file was submitted

SUBBJP determines if a job file is to be submitted, then causes itself to roll out until another file is to be submitted. A job file is submitted if the difference between the last time it was scheduled and the current time exceed the periodic interval specified. The last time is then updated.

Format

SUBBJP
BDF=filename

Parameters

BDF

Specifies the name of the batch directive file that contains records defining the job file(s) to be submitted periodically and the intervals between job submissions. The batch directive file must be indirect access and resident in the user's catalog.

Each batch directive file record consists of eight parameters which must be specified on one line and separated by spaces. These parameters and their required order are as follows:

BDF Record Parameters	Description
File name	Specifies the name of the job file to be submitted.
User name	Specifies the user name of the job file to be submitted.
Submit date	Six digits specifying the year, month, and day to begin job file submission. Format: yymmdd.
Submit time	Four digits specifying the hour and minute of the submit date to begin job file submission. Format: hhmm.
Skip day	Two digits specifying the number of days to skip. Format: dd.
Skip time	Four digits specifying the number of hours and minutes to skip. Format: hhmm.
Last date run	Six digits specifying the year, month, and day on which the job file was last submitted. Format: yymmdd.
Last time run	Four digits specifying the hour and minute that the job file was last submitted. Format: hhmm.

Examples

1. In this example batch directive file record, file NPAPERD with user name NETNPA is to be submitted for execution on a daily basis at 01:10, beginning on the first of January, 1985.

```
NPAPERD NETNPA 850101 0110 01 0000 850101 0110
```

2. In this example batch directive file record, file NPAPERD with user name NETNPA is to be submitted for execution on a weekly basis, once every seven days, beginning at 04:35 on January 1, 1985.

```
NPAPERD NETNPA 850101 0435 07 0000 850101 0435
```

The following four examples show how to facilitate the generation of the necessary files for this procedure:

1. Batch job directive file (NPABDF).

```
NPABT1 NETADMN 850101 0600 01 0000  850101 0600
NPABT2 NETADMN 850101 0600 07 0005  850101 0600
NPABT3 NETADMN 850101 0000 00 0100  850101 0000
```

2. Batch job for creating the hourly reports at the end of the day (NPABT1).

```
NETNPA.
USER,NETADMN,NETNPA.
CHARGE,*.
NPA.
CRECAR,RN=HOURLY,BDO=1,EDO=1.
REWIND,*.
COPYEI,CREOUT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
EXIT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
```

3. Batch job for creating daily reports at the end of the week (NPABT2).

```
NETNPA.
USER,NETADMN,NETNPA.
CHARGE,*.
NPA.
CRECAR,RN=DAILY,BDO=7,EDO=1.
REWIND,*.
COPYEI,CREOUT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
EXIT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
```

4. Batch job for causing the reformat process to be run every hour (NPABT3).

```
NETNPA.
USER,NETADMN,NETNPA.
CHARGE,*.
NPA.
REFCLF.DB=ALL
REWIND,*.
COPYEI,REFOUT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
EXIT.
DAYFILE,PL=60.
ROUTE,OUTPUT,DC=PR.
```

Aborts and Error Recovery

The CCL interface responses to incorrect values are utilized for checking input parameters. The CCL procedure takes an abort exit if the application fails. This exit indicates that the application has failed. Responses from the normal system job commands indicate the reason for failure.

DI Dump Analyzer Commands 19

DI Dump Analyzer Commands 19

The DI Dump Analyzer is used to reformat the file produced when a DI system dumps its memory on reset. In the default environment, DIs dump their memory whenever they are reset, with two exceptions: 1) When the DI is reset manually, with the power switch, and 2) When the DI is reset by NETOU's KILL_SYSTEM (KILS) command, with the DUMP parameter set to NO.

The DI Dump Analyzer program runs on the host, where DI dump files are maintained.

If you are using the Dump Analyzer under NOS, you may begin a session at any time by using the ANACD command.

NOTE

To use the Dump Analyzer under NOS/VE, you must first use the following CREATE_COMMAND_LIST_ENTRY (CRECLE) command:

```
CRECLE $SYSTEM.CDCNET.VERSION_INDEPENDENT.COMMAND_LIBRARY
```

The CRECLE command makes the Dump Analyzer available to your NOS/VE job. This command may be added to your user prolog, so that the DI Dump Analyzer is available anytime you log onto the host. Once the Dump Analyzer is available to your NOS/VE job, you may begin a Dump Analyzer session with the ANACD command.

The ANACD command specifies the dump file to be analyzed, the version level of the dump, input and output files, and the STATUS parameter (effective only under NOS/VE). After you enter the ANACD command on NOS/VE or NOS, the DI Dump Analyzer is ready to accept any of the following subcommands:

```
DISPLAY_AUTO_DUMP_TABLE
DISPLAY_BUFFER_CHAIN
DISPLAY_COMMAND_INFORMATION
DISPLAY_DATA_QUEUE
DISPLAY_EXECUTIVE_ERROR_TABLE
DISPLAY_LINKED_LIST
DISPLAY_LOG_QUEUES
DISPLAY_MEMORY_HEADER
DISPLAY_MEMORY_USERS
DISPLAY_PROGRAM_VALUE
DISPLAY_TASK_CONTROL_BLOCK
HELP
SELECT_TASK
VALIDATE_STACK_AREAS
DISPLAY_BOARD_MAP_TABLE
DISPLAY_CALLS
DISPLAY_COMMAND_LIST
DISPLAY_DI_SYSTEM_STATUS
DISPLAY_HARDWARE_STATUS
DISPLAY_LINE_CONTROL_BLOCK
DISPLAY_MEMORY
DISPLAY_MEMORY_MAP
DISPLAY_NETWORK_STATUS
DISPLAY_SYSTEM_CONFIG_TABLE
DISPLAY_TREE
QUIT
VALIDATE_GLOBAL_INFORMATION
```

ANALYZE_CDCNET_DUMP (ANACD)

Purpose The ANALYZE_CDCNET_DUMP command begins a Dump Analyzer session on the host.

Format

ANALYZE_CDCNET_DUMP
DUMP_FILE = dump_file_name
VERSION = dump_file_version_level
INPUT = input_file_name
OUTPUT = output_file_name
STATUS = status_variable_name

Parameters *DUMP_FILE (DF)*

This parameter specifies the name of the file containing the CDCNET DI dump to be analyzed.

If the DUMP_FILE parameter is omitted, file name DUMPFIL is assumed.

VERSION (V)

This parameter specifies the version level of the Dump Analyzer to be used for analyzing the dump file. No radix can be specified on this parameter, it is simply entered as four hexadecimal digits.

If the VERSION parameter is omitted, the Dump Analyzer version used is the one selected by the site administrator, usually the version level of the CDCNET software currently installed at the site.

INPUT (I)

This parameter specifies the name of the file (ASCII 6/12 on NOS) containing Dump Analyzer subcommands in SCL format. Under NOS/VE, this file is opened at the beginning-of-information (BOI) by default. Under NOS, this file must be local, and it is not rewound before it is read.

If the INPUT parameter is omitted, file name $COMMAND (INPUT on NOS) is assumed. This file is usually attached to the terminal.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE, and should include file position if you want other than the default position, beginning-of-information (BOI). This file is local on NOS, where it is always positioned at the end-of-information (EOI). Output is written in ASCII format.

During the Dump Analyzer session, output can be redirected for individual subcommands that have their own OUTPUT parameters.

If the OUTPUT parameter is omitted, file name $OUTPUT (OUTPUT on NOS) is assumed. This file is usually attached to the terminal.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is ignored.

Remarks

- ANALYZE_CDCNET_DUMP is abbreviated ANACD. Under NOS/VE, either the full or abbreviated forms of the ANALYZE_CDCNET_DUMP command and its parameters may be used to begin a Dump Analyzer session. Only the abbreviated form of the command, and its parameters, may be used under NOS.

- Under NOS/VE, Dump Analyzer error messages are always written to the file $ERRORS, which is usually attached to your terminal. Under NOS, Dump Analyzer messages are always written to file OUTPUT. Under both operating systems, error messages are also written to the currently active output file, if this is not a terminal.

  NOTE

  The file named in an output parameter on a subcommand takes precedence over the output file named on the ANACD command during execution of that subcommand.

Examples

The following NOS command accepts subcommands from file CMDFILE and writes output to the terminal.

```
ANACD,DF=DIAA132,I=CMDFILE
```

The following NOS command begins an interactive Dump Analyzer session, specifying 2404 as the version of the Dump Analyzer to be used.

```
ANACD,DF=DSA9189,V=2404
```

The following NOS/VE command begins an interactive Dump Analyzer session on a dump file from SYSTEM_0800253000DA.

```
ANACD,DF=$SYSTEM.CDCNET.DUMP.SYSTEM_0800253000DA.FULL_8701050856249189
```

Initial output from the ANACD command contains the Dump Analyzer banner message, and is formatted as follows:

```
COPYRIGHT CONTROL DATA SYSTEMS, INC. 1985, 1992
ALL RIGHTS RESERVED

CDCNET DUMP ANALYZER VERSION = 1614

CDCNET DI SOFTWARE BOOT VERSION RECORDED IN MPB_RAM = 1614

CDCNET DI SOFTWARE RELEASE LEVEL RECORDED IN SYSTEM_DATA = 1614


The reset_code found in the dump file is
          18(16) = TASK_ERROR_NO_RECOVERY_PROC
```

NOTE

In order to function properly, the Dump Analyzer must be able to locate and interpret certain data structures from the DI dump file. Because these data structures can change (in location or structure) from version to version, it is important to use a Dump Analyzer that can read the dump file under analysis. If you do not specify an alternative version of the Dump Analyzer, the version used is the version selected by your CDCNET site administrator.

To help identify which alternative Dump Analyzer version should be used, two version numbers are displayed at the start of the Dump Analyzer session. One identifies the official **release level** of the CDCNET software product. This version number is stored into the DI's System Data Table during initialization.

The other version number, the **boot version**, is stored in MPB RAM and identifies the software version of the boot file that is used to reload the DI. Unless your site develops software in conjunction with Control Data, the boot version number should match the release level version number.

If the version level of the Dump Analyzer you are using does not match the release level version of the dump file, the Dump Analyzer displays diagnostic message number 86. You might find it necessary to restart the Dump Analyzer program and specify the CDCNET software version level of the dump file for the VERSION (V) parameter on the ANACD command. If a copy of the Dump Analyzer program built at this software version level is available at the site, it is used.

Summary of Subcommands

Subcommands processed by the Dump Analyzer fall into six major categories, as summarized in the following table.

Dump Analyzer Control Subcommand	Description
QUIT (QUI)	Terminates the Dump Analyzer session and returns control to the operating system.

Dump Analyzer Help Subcommands	Description
DISPLAY_COMMAND_INFORMATION (DISCI)	Displays parameter information for a specified Dump Analyzer subcommand.
DISPLAY_COMMAND_LIST (DISCL)	Displays a list of available Dump Analyzer subcommands.
HELP (HEL)	Displays a list of available Dump Analyzer subcommands, or parameter information for a specified Dump Analyzer subcommand.

Dump Summary Subcommands	Description
DISPLAY_AUTO_DUMP_TABLE (DISADT)	Displays the contents of the auto dump table and the map of memory available in the dump file.
DISPLAY_BOARD_MAP_TABLE (DISBMT)	Displays the contents of the board map table.
DISPLAY_DI_SYSTEM_STATUS (DISDSS)	Displays system configuration information from the time of reset.
DISPLAY_EXECUTIVE_ERROR_TABLE (DISEET)	Displays information from the executive error table.
DISPLAY_HARDWARE_STATUS (DISHS)	Displays information about the modular DI hardware from the time of reset.
DISPLAY_ICA_SYSTEM_STATUS (DISISS)	Displays system configuration information from the time of reset.
DISPLAY_NETWORK_STATUS (DISNS)	Displays status of networks connected to the DI at reset.
DISPLAY_SYSTEM_CONFIG_TABLE (DISSCT)	Displays information from the system configuration table.
VALIDATE_GLOBAL_INFORMATION (VALGI)	Displays general diagnostic information, serving as a preliminary guide to further analysis.

Address-Oriented Subcommands	Description
DISPLAY_BUFFER_CHAIN (DISBC)	Displays buffer chain information starting from the specified machine address. All descriptor buffers or data buffers are displayed for all messages in the chain.
DISPLAY_DATA_QUEUE (DISDQ)	Displays buffer chain information associated with the queue control block at the specified machine address.
DISPLAY_LINKED_LIST (DISLL)	Displays a linked list of structured elements given the machine address of the first element and offset to a pointer within the structure linking it to the next item in the list.
DISPLAY_MEMORY (DISM)	Displays memory contents. The display begins at a specified machine address or entry-point address and is of a specified length. Both hexadecimal and ASCII formats are displayed.
DISPLAY_MEMORY_HEADER (DISMH)	Displays information from the allocation header of the memory extent that contains the specified address.
DISPLAY_TREE (DIST)	Displays all nodes of a binary tree structure, or a specified node matching a user-supplied key.

Task-Oriented Subcommands	Description
DISPLAY_CALL (DISC)	Displays information about the dynamic call chain (or module traceback) of the specified task, or all tasks.
DISPLAY_MEMORY_USERS (DISMU)	Displays information about the allocation of memory extents to various tasks or other users in the system.
DISPLAY_TASK_CONTROL_BLOCK (DISTCB)	Displays information from the task control block for the specified task, or all tasks.
SELECT_TASK (SELT)	Specifies the task to be examined in subsequent task-oriented subcommands (designates the task to be considered the current task).
VALIDATE_STACK_AREAS (VALSA)	Checks integrity of the reserved and user stack areas of any or all tasks.

Miscellaneous Subcommands	Description
DISPLAY_LINE_CONTROL_BLOCK (DISLCB)	Displays information from the configured line control block for a given line, or all lines.
DISPLAY_LOG_QUEUES (DISLQ)	Displays the preserve and initialization log message queues.
DISPLAY_MEMORY_MAP (DISMM)	Displays a memory map of the modules that were loaded in the DI before it was reset.

Subcommand Syntax and Conventions

Following are descriptions of syntax and conventions for Dump Analyzer subcommands.

Subcommand Entry

- Subcommands and parameters may be specified either in their full or abbreviated forms on NOS/VE or NOS.
- Under NOS/VE, Dump Analyzer subcommands can be continued from one line to the next according to standard SCL conventions. Under NOS, the Dump Analyzer processes exactly one subcommand per line.
- Subcommand names are formed with a verb followed by an object (for example, DISPLAY_BUFFER_CHAIN, where DISPLAY is the verb and BUFFER_CHAIN is the object).
- Subcommand names may be abbreviated to the first three characters of the verb followed by the first character of each word in the object (for example, DISPLAY_DATA_QUEUE is abbreviated DISDQ).

Parameter Specification

- Subcommand parameters may be abbreviated to the first letter of each word (for example, DISPLAY_OPTION is abbreviated DO). However, the STATUS parameter (used under NOS/VE) must not be abbreviated.
- Subcommand parameters may be specified by equating the parameter name with the desired value (such as DO=E), or they may be specified positionally, without the parameter name.

  If parameters are specified positionally, commas must be used to skip over omitted parameters (for example, DISM 256,,16,4, where the second parameter is omitted). Otherwise, subcommand parameters may be separated by commas or spaces.

Numeric Values

- Numeric values for parameters are of the form nn(ra), where nn is a string of digits and ra is the radix. Valid radixes are 2, 8, 10, and 16. *It is true for all numeric parameters* that if no radix is specified, the radix 16 is assumed.

  Hexadecimal values starting with digits A through F must be preceded by zero, making them distinct from literals. (For example, 0FFF(16) is interpreted as a numeric, but FFF(16) is interpreted as a literal.)
- Integer values specified on parameters must be greater than zero, unless otherwise indicated in the subcommand format.

Address Values

- Numerically expressed addresses must be integers in the range 0..7FFFFFFF(16). See the individual subcommands for alternatives to numeric expression. (For example, some parameters allow you to specify externally defined entry points by name, as in ADDRESS=major_card_status_table.)

File Name Values

- File names specified on parameters may be up to 31 characters long under NOS/VE or NOS, but only the first seven characters are used to determine the identity of the file if the Dump Analyzer is running under NOS. This permits you to use unchanged procedures or directives files on either operating system, even if they contain subcommands with long file names. The Dump Analyzer does not ensure that a long file name shortened for NOS is unique.

  Under NOS/VE, file names may begin with, and include anywhere in the name, the alphabetic characters A through Z or any of the four special characters in the set ($, #, @, and _). NOS/VE file names may also include, but must not begin with, numerals. File names containing the special characters ($, #, @, and _) are only legal under NOS/VE.

  Under NOS, file names must begin with an alphabetic character, A through Z, but may include numerals.

Boolean Values

- Boolean parameters can be specified by any of their standard aliases. These are: TRUE/FALSE, YES/NO, and ON/OFF.

STATUS Parameter

- The STATUS parameter is operational only under NOS/VE. It is used to report errors. For a complete description of the STATUS parameter, see the NOS/VE System Usage manual.

  Under NOS, the STATUS parameter is checked for correct syntax (it must be a valid NOS name), but output is not otherwise affected by its use.

DISPLAY_AUTO_DUMP_TABLE (DISADT) ANACD Subcommand

Purpose This subcommand displays the contents of the auto dump table. This display identifies the DI address ranges available in the dump file.

Format

DISPLAY_AUTO_DUMP_TABLE
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This optional parameter specifies the name of the file on which the information is to be written. If the output file name is not specified, information is written to the output file specified on the ANACD control statement.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The DISADT subcommand shown below, produces an auto dump table display with the information written on the output file named ADT.

```
DISADT O=ADT
```

Output from this subcommand is formatted as follows:

```
AUTO DUMP TABLE DISPLAY

Auto Dump Table Checksum   =     0D7E(16)
Auto Dump Table Address    =   10B29E(16)
Auto Dump Table Byte Count =      35A(16)
```

Entry Number	Start Address	End Address	Number of Bytes
1	0	678	678
2	88E	2EB4	2626
3	2F72	4000	108E
4	90000	96000	6000
5	0A0000	0A8000	8000
6	100000	10CAF0	0CAF0
7	10D5A0	10D930	390
8	10D97C	10DF92	616
9	111ECE	114696	27C8
10	116C70	117C2C	0FBC
11	117DA8	117FBC	214
12	119C88	11A67A	9F2
13	11CA66	11D9D6	0F70
14	11ED22	11FF18	11F6
15	11FFFC	12019E	1A2
16	12025C	120508	2AC
17	1210E8	121322	23A
18	12362A	124E82	1858
19	124EA8	125828	980
20	125C9C	125E8A	1EE
.	.	.	.
.	.	.	.
.	.	.	.
n	nnnnnn	nnnnnn	nnnn

where n is the total number of ADT entries.

DISPLAY_BOARD_MAP_TABLE (DISBMT) ANACD Subcommand

Purpose This subcommand displays selected fields of the board map table. This allows users to determine if a reset has occurred as a result of a hardware problem.

Format

DISPLAY_BOARD_MAP_TABLE
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This optional parameter specifies the name of the file on which the information is to be written. If the output file name is not specified, information is written to the output file specified on the ANACD control statement.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks If there are no error counts, then the field is left blank. The error type is displayed in the Error Description column (see the following example).

Failing addresses are displayed in hexadecimal. All other integer values are displayed in decimal.

If a slot is empty, the line for that slot is left blank.

Examples The DISBMT subcommand shown below, produces a board map table with the information written on the output file named MAP.

```
DISBMT O=MAP
```

Output from this subcommand is formatted as follows:

```
BOARD MAP TABLE DISPLAY

Conditions shown exist after reset

      Card   Card              Failing
Slot  Type   OK     Status     Address     Counts  Error Description
-------------------------------------------------------------------

  0   MPB    Yes                100AC        3    Fatal Error
  1   PMM    Yes                15AC73      17    Parity Error
  2   SMM    Yes                200000       5    Single-Bit Error
                                             2    Multi-Bit Error
  3   ESCI   Yes
  4
  5
  6   ESCI   No                                   Transceiver Bad
  7   CIM    Yes

LIM                                              Parent CIM
Slot  LIM Type           LIM Status              Slot
-----------------------------------------------------------

  0   RS232 8-Channel    Not tested              $CIM 7
  1   RS232              OK                        "
  2   URI                Not tested                "
  3   X.24               OK                        "
  4   V.35               Bad                       "
  5   UNKNOWN                                      "
  6                                                "
  7   RS449              OK                        "
```

NOTE

In the case of MPA RAM, the failing address is biased by 8000(16).

DISPLAY_BUFFER_CHAIN (DISBC) ANACD Subcommand

Purpose This subcommand displays information from selected buffer chains starting from the specified machine address. Descriptor buffers are summarized and, depending on the display option selected, data from the associated data buffers is displayed.

Format

DISPLAY_BUFFER_CHAIN
ADDRESS = address or entry point name
DISPLAY_OPTION = keyword
REPEAT_COUNT = integer < 4096 or *ALL*
OUTPUT = file_name
STATUS = status_variable_name

Parameters

ADDRESS (A)

This required parameter specifies the starting address of the buffer chain to be displayed. The value may be a memory address or an entry point name (externally defined variable) whose address is used as the base address for the display. Entry point names are found by using the DISPLAY_MEMORY_MAP subcommand described in this chapter.

DISPLAY_OPTION (DO)

This parameter specifies the amount of data to be displayed. If FULL (or EXPANDED) is specified, the descriptors and data for all messages starting from the address specified to the end of the message chain are displayed. If BRIEF (or SUMMARY) is specified, only summary information indicating the starting address of the descriptor for each message chain, the number of buffers chained for each message, and the total characters in the message are displayed.

If the DISPLAY_OPTION parameter is not specified, FULL is assumed.

REPEAT_COUNT (RC)

This parameter indicates the number of messages to be displayed. If the value specified is greater than the number of message chains encountered, all the message chains from the specified starting descriptor are displayed. The keyword ALL also causes all the message chains from the specified starting descriptor to be displayed (up to a maximum of 4096 messages).

If REPEAT_COUNT is not specified, all the message chains from the specified starting descriptor are displayed (up to a maximum of 4096 messages).

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks The data enclosed in brackets in the data buffer display is the area in use (see the example below). Data following the area in use is not displayed.

Examples The DISPLAY_BUFFER_CHAIN subcommand, shown below, writes information from the buffer chain beginning at address 19188E(16) to the ANACD output file.

```
DISPLAY_BUFFER_CHAIN ADDRESS=19188E
```

Output from this subcommand is formatted as follows:

```
BUFFER CHAIN DISPLAY


DATA DESCRIPTOR       19188E(16)     NEXT_DESCRIPTOR       1A5422(16)
NEXT_MESSAGE              0(16)      THE_DATA              18F3E4(16)
OFFSET                   5B(16)      COUNT_BUFFER             36(16)
COUNT_MESSAGE           0BD(16)      USAGE_DESCRIPTOR       FFFF(16)

USAGE COUNT            FFFF(16)
DATA AT THE_DATA:

  18F3E4  FFFF  4552  464C  4F57  4F56  4552  464C  4F57     ERFLOWOVERFLOW
  18F3F4  4F56  4552  464C  4F57  4F56  4552  464C  4F57   OVERFLOWOVERFLOW
  18F404  4F56  4552  464C  4F57  4F56  4552  464C  4F57   OVERFLOWOVERFLOW
  18F414  4F56  4552  464C  4F57  4F56  4552  464C  4F57   OVERFLOWOVERFLOW
  18F424  4F56  4552  464C  4F57  4F56  4552  464C  4F57   OVERFLOWOVERFLOW
  18F434  5010  132F  42C9  004E  0001  0008 [3501  0185   P  /B  N     5
  18F444  0114  1337  2904  3000  0000  0108  0025  1000      7) 0       %
  18F454  7300  1154  4449  3131  3520  2020  2020  2020   s  TDI115
  18F464  2020  2020  2020  2020  2020  2020  2020  2020
  18F474  2020]


DATA DESCRIPTOR       1A5422(16)     NEXT_DESCRIPTOR       1A5896(16)
NEXT_MESSAGE              0(16)      THE_DATA              121170(16)
OFFSET                    1(16)      COUNT_BUFFER             7B(16)
COUNT_MESSAGE            87(16)      USAGE_DESCRIPTOR       FFFF(16)

USAGE COUNT            FFFF(16)
DATA AT THE_DATA:

  121170  FFFF [0B01  01DA  0B01  01E0  0B01  13A5  0A03
  121180  0000  0001  0A05  0800  2530  011B  0B01  01E1          %0
  121190  0A01  0010  124E  6465  666C  2C20  332C  2033       Ndefl, 3, 3
  1211A0  2C20  6173  796E  6374  6970  2C20  6C69  6E65   , asynctip, line
  1211B0  3333  2C20  4152  3D53  4350  2C20  6C74  3D73   33, AR=SCP, lt=s
  1211C0  7769  7463  6865  642C  2063  6374  3D33  3630   witched, cct=360
  1211D0  2C20  6364  743D  3132  302C  2074  7570  3D74   , cdt=120, tup=t
  1211E0  7570  7063  720B  0101  E20A  0180  EC]          uppcr
                  .                   .
                  .                   .
                  .                   .
```

The following DISPLAY_BUFFER_CHAIN subcommand displays summary information for the buffer chains starting at address 18A0F0(16).

```
DISBC A=18A0F0 DO=B
```

Output from this subcommand is formatted as follows:

```
BUFFER CHAIN DISPLAY

          STARTING ADDRESS    NUMBER OF    TOTAL CHARACTERS PER MESSAGE
           OF DESCRIPTOR       BUFFERS           (COUNT_MESSAGE)

MESSAGE # 1     18A0F0(16)          1          2A(16)=      42(10)
MESSAGE # 2     192539(16)        2B9        8E86(16)=   36486(10)
MESSAGE # 3     1A37C8(16)        12B       3547BC(16)=  491772(10)
```

If you want, the DISPLAY_BUFFER_CHAIN subcommand could next be used to display the data in the buffer chains for particular messages by using the addresses given above, a REPEAT_COUNT restriction, and the DO=FULL option.

Message chains that contain zero byte buffers or invalid buffer_count data are flagged with an alert following the message summary line, as in the following example:

```
   ...
MESSAGE # 7     1E00A2(16)          1C         24F(16)=     1103(10)
     ****    ZERO BYTE BUFFER ENCOUNTERED IN PRECEDING MESSAGE
   ...
```

DISPLAY_CALLS (DISC)
ANACD Subcommand

Purpose

The DISPLAY_CALL subcommand displays information about nested procedure calls for a specified task or for all tasks in the dump file. The display is ordered from the most recent call to the least recent call. The module name and the name of the nearest entry point preceding the address of interest in that module are displayed.

Format

DISPLAY_CALL
TASK_IDENTIFIER = address, task_name or *keyword*
REPEAT_COUNT or *COUNT = integer 1..4096* or *keyword*
START = integer 1..4096
STACK_FRAME = boolean
OUTPUT = file_name
STATUS = status_variable_name

Parameters

TASK_IDENTIFIER (TI)

This parameter specifies the Task Control Block (TCB) address for the task to be examined. If a task name is specified, call information is displayed for all the TCBs scheduled for that task. The keyword ALL causes all tasks to be processed, beginning with the system ancestor task.

If TASK_IDENTIFIER is omitted, only the currently selected task is processed. The currently selected task is the task specified on the SELECT_TASK subcommand. If SELECT_TASK has not been used to specify a task, the task running when the DI was reset is assumed. If no task was running when the DI was reset, the Dump Analyzer issues diagnostic message number 37.

REPEAT_COUNT (RC) or *COUNT (C)*

This parameter specifies the number of calls to be displayed. If the keyword ALL is specified, or if REPEAT_COUNT is greater than the number of existing calls, all calls from the specified starting call are displayed (up to a maximum of 4096 calls).

If COUNT is omitted, all calls from the specified starting call are displayed (up to a maximum of 4096 calls).

START (S)

This parameter specifies which call on the chain is to be displayed first. This parameter makes it possible to eliminate display of the most recent calls. The value 1 represents the most recent call, 2 represents its predecessor, and so forth. For example, S=5 causes the display to begin with the fifth most recent call.

If START is omitted, display begins with the most recent call.

STACK_FRAME (SF)

This parameter specifies whether the stack frame(s) associated with the call should be displayed. Addresses within a stack frame display are expressed as offsets from the stack frame's starting address. For SF=TRUE (ON or YES), all stack frames are displayed; for SF=FALSE (OFF or NO), no stack frames are displayed.

If STACK_FRAME is omitted, no stack frames are displayed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks If module and entry point information is not available in the dump file, only the absolute address for the specified call or calls is displayed.

Examples The following subcommand displays five calls, starting with the third most recent call in the call chain for the currently selected task.

```
DISPLAY_CALL START=3 REPEAT_COUNT=5
```

or

```
DISC S=3 RC=5
```

Output from the DISPLAY_CALL subcommand is displayed in one of the following three formats.

1. If module and entry point information is available:

```
TCB ADDRESS              1044D6(16)

   TRACEBACK FROM MODULE  EXEC_PMM + 142(16)
    NEAREST PRECEDING ENTRY POINT   CALL_SURE_BG

   CALLED FROM  - MODULE  SYSTEM_ANCESTOR + 1FCC(16)
    NEAREST PRECEDING ENTRY POINT   RESET_DI

   CALLED FROM  - MODULE  SYSTEM_ANCESTOR + 458(16)
    NEAREST PRECEDING ENTRY POINT   SYSTEM_ANCESTOR
```

2. If module and entry point information is not available:

```
TCB ADDRESS              13BA96(16)

    Traceback from address      100FC6(16)
    Called from  - address      101234(16)
    Called from  - address      1010C8(16)
```

3. If stack frame display is requested (that is, SF=TRUE):

```
TCB ADDRESS            139244(16)

  TRACEBACK FROM MODULE  EXEC_PMM + 142(16)
   NEAREST PRECEDING ENTRY POINT   CALL_SURE_BG

 STACK FRAME STARTING AT    1BD06(16) AND ENDING AT    1BD91(16) :

        0 1  2 3  4 5  6 7  8 9  A B  C D  E F
        -----------------------------------------
 1BD06  0001 BD92 0015 395C 0000 0052 0008 0002        9\   R
 1BD16  3135 3A30 323A 3533 5354 4143 0008 0002   15:02:53STAC
 1BD26  3131 2F32 312F 3836 5354 4143 4B20 00E4   11/21/86STACK
 1BD36  5354 0000 0027 2A20 4449 2052 6573 6574   ST   '* DI Reset
 1BD46  2C20 536F 6674 7761 7265 2045 7272 6F72   , Software Error
 1BD56  2043 6F64 6520 3D20 2033 3220 2A20 2020    Code =  32 *
 1BD66  2020 2020 2020 2020 2020 2020 2020 2020
 1BD76  2020 2020 2020 2020 2020 2020 2020 2020
 1BD86  2020 2020 20DE FF01 BDA4 0001


  CALLED FROM  - MODULE  SYSTEM_ANCESTOR + 222C(16)
   NEAREST PRECEDING ENTRY POINT   RESET_DI

 STACK FRAME STARTING AT    1BD92(16) AND ENDING AT    1BDA3(16) :

        0 1  2 3  4 5  6 7  8 9  A B  C D  E F
        -----------------------------------------
 1BD92  0001 BDA4 0001 2596 0000 0032 FF83 0024        %    2   $
 1BDA2  9864                                      d


  CALLED FROM  - MODULE  COMMON_CYBIL_SUBROUTINES_PMM + 2AA(16)
   NEAREST PRECEDING ENTRY POINT   ABORT_SYSTEM

 STACK FRAME STARTING AT    1BDA4(16) AND ENDING AT    1BDB9(16) :

        0 1  2 3  4 5  6 7  8 9  A B  C D  E F
        -----------------------------------------
 1BDA4  0001 BDBA 0027 85EA 0000 0032 001F B9DA        '      2
 1BDB4  001F 985E 0024                                ^ $


  CALLED FROM  - MODULE  EMAL_TEST + 32(16)

 STACK FRAME STARTING AT    1BDBA(16) AND ENDING AT    1BDF9(16) :

        0 1  2 3  4 5  6 7  8 9  A B  C D  E F
        -----------------------------------------
 1BDBA  0001 BDFA 0027 9D4C 001F B9DA 001F 0011        ' L
 1BDCA  152C 001F B4EA 4143 4B20 00F8 5354 0024     ,    ACK   ST $
 1BDDA  985E 0024 9864 0024 97CC 0024 9864 0043     ^ $ d $   $ d C
 1BDEA  0120 00FB 0024 97C6 0000 0000 5354 4143       $       STAC


  CALLED FROM  - MODULE  EMAL_TEST + 1794(16)
   NEAREST PRECEDING ENTRY POINT   EMAL_7

 STACK FRAME STARTING AT    1BDFA(16) AND ENDING AT    1BDFD(16) :

        0 1  2 3  4 5  6 7  8 9  A B  C D  E F
        -----------------------------------------
 1BDFA  0001 BDFE


  CALLED FROM  - MODULE  OLL_PROGRAM_INTERFACE_PROCS + 171A(16)
   NEAREST PRECEDING ENTRY POINT   START_NAMED_TASK_AND_PROCEED
```

DISPLAY_COMMAND_INFORMATION (DISCI) ANACD Subcommand

Purpose

The DISPLAY_COMMAND_INFORMATION subcommand displays a list of parameters associated with a given Dump Analyzer subcommand, including types and default values. This subcommand displays information in the same format as the CDCNET and NOS/VE commands of the same name.

NOTE

The description below applies only to DISPLAY_COMMAND_INFORMATION on NOS. If you are using the Dump Analyzer under NOS/VE, the SCL DISPLAY_COMMAND_INFORMATION command supplies similar information. (See the NOS/VE System Usage manual.)

Format

DISPLAY_COMMAND_INFORMATION
COMMAND_NAME=name
OUTPUT=file_name
STATUS=status_variable_name

Parameters

COMMAND_NAME (CN)

This parameter specifies the name of the Dump Analyzer subcommand for which parameter information is to be displayed. The value may be the full name of a Dump Analyzer subcommand, or any of its valid abbreviations.

This parameter is required. If the specified subcommand name cannot be found, a warning is issued with no further display.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid NOS file name. This file is local.

If OUTPUT is not specified, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples

The following subcommand displays subcommand information for the DISPLAY_CALL subcommand:

```
DISCI DISC
```

Output from this subcommand is formatted as follows:

```
TASK_IDENTIFIER, TID, TI    : any = $optional
COUNT, C, REPEAT_COUNT, RC : integer  1 .. 0FFF(16) or key ALL = ALL
START, S                    : integer  1 .. 0FFF(16) = 1
STACK_FRAME, SF             : boolean = FALSE
OUTPUT, O                   : name = $optional
STATUS                      : name = $optional
```

DISPLAY_COMMAND _LIST (DISCL) ANACD Subcommand

Purpose The DISPLAY_COMMAND_LIST subcommand displays the list of subcommands that are currently available for execution by the Dump Analyzer. This command is patterned after the CDCNET command by the same name.

NOTE

The description below applies only to DISPLAY_COMMAND_LIST on NOS. If you are using the Dump Analyzer under NOS/VE, use the SCL DISPLAY_COMMAND_LIST_ENTRY command for similar information. (See the NOS/VE System Usage manual.) There are differences in the output format between the Dump Analyzer's DISCL command and the SCL DISCLE command.

Format

DISPLAY_COMMAND _LIST
DISPLAY_OPTION=keyword
OUTPUT=file_name
STATUS=status_variable_name

Parameters *DISPLAY_OPTION (DO)*

This parameter specifies the detail of information to be displayed. If the keyword FULL (F, EXPANDED, E, or ALL) is specified, all commands are listed both by their complete names and by any abbreviated forms. If the keyword BRIEF (B, SUMMARY, or S) is specified, then a list is displayed that gives only the full name of each subcommand.

If the DISPLAY_OPTION parameter is not specified, subcommand names are displayed in the BRIEF format.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid NOS file name. This file is local.

If OUTPUT is not specified, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks The HELP subcommand, with no parameters, results in the same output as display_command_list do=b.

Examples The following subcommands display a list of the available Dump Analyzer subcommands in BRIEF mode:

or DISCL

```
DISCL DO=B
```

Output from this subcommand is formatted as follows:

```
ANACD COMMAND LIST
------------------

DISPLAY_AUTO_DUMP_TABLE                DISPLAY_BOARD_MAP_TABLE
DISPLAY_BUFFER_CHAIN                   DISPLAY_CALLS
DISPLAY_COMMAND_INFORMATION            DISPLAY_COMMAND_LIST
DISPLAY_DATA_QUEUE                     DISPLAY_DI_SYSTEM_STATUS
DISPLAY_EXECUTIVE_ERROR_TABLE          DISPLAY_HARDWARE_STATUS
DISPLAY_LINKED_LIST                    DISPLAY_LINE_CONTROL_BLOCK
DISPLAY_LOG_QUEUES                     DISPLAY_MEMORY
DISPLAY_MEMORY_HEADER                  DISPLAY_MEMORY_MAP
DISPLAY_MEMORY_USERS                   DISPLAY_NETWORK_STATUS
DISPLAY_PROGRAM_VALUE                  DISPLAY_SYSTEM_CONFIG_TABLE
DISPLAY_TASK_CONTROL_BLOCK             DISPLAY_TREE
HELP                                   QUIT
SELECT_TASK                            VALIDATE_GLOBAL_INFORMATION
VALIDATE_STACK_AREAS
```

The following subcommand displays a list of the Dump Analyzer subcommands in FULL format.

```
discl do=f
```

Output from this subcommand is formatted as follows:

```
ANACD COMMAND LIST
------------------

DISPLAY_AUTO_DUMP_TABLE                DISADT
DISPLAY_BOARD_MAP_TABLE                DISBMT
DISPLAY_BUFFER_CHAIN                   DISBC
DISPLAY_CALLS                          DISPLAY_CALL,DISC
DISPLAY_COMMAND_INFORMATION            DISCI
DISPLAY_COMMAND_LIST                   DISCL
DISPLAY_DATA_QUEUE                     DISDQ
DISPLAY_DI_SYSTEM_STATUS               DISDSS
DISPLAY_EXECUTIVE_ERROR_TABLE          DISEET
DISPLAY_HARDWARE_STATUS                DISHS
DISPLAY_LINKED_LIST                    DISLL
DISPLAY_LINE_CONTROL_BLOCK             DISLCB
DISPLAY_LOG_QUEUES                     DISLQ
DISPLAY_MEMORY                         DISM
DISPLAY_MEMORY_HEADER                  DISMH
DISPLAY_MEMORY_MAP                     DISMM
DISPLAY_MEMORY_USERS                   DISMU
DISPLAY_NETWORK_STATUS                 DISNS
DISPLAY_PROGRAM_VALUE                  DISPV
DISPLAY_SYSTEM_CONFIG_TABLE            DISSCT
DISPLAY_TASK_CONTROL_BLOCK             DISTCB
DISPLAY_TREE                           DIST
HELP                                   HEL
QUIT                                   QUI
SELECT_TASK                            SELT
VALIDATE_GLOBAL_INFORMATION            VALGI
VALIDATE_STACK_AREAS                   VALSA
```

DISPLAY_DATA_QUEUE (DISDQ) ANACD Subcommand

Purpose The DISPLAY_DATA_QUEUE subcommand displays information about the specified queue control block (QCB) and the buffer chains to which it points.

Format

DISPLAY_DATA_QUEUE
ADDRESS = address or **entry point name**
DISPLAY_OPTION = keyword
REPEAT_COUNT = integer 1..4096 or *keyword*
OUTPUT = file_name
STATUS = status_variable_name

Parameters

ADDRESS (A)

This required parameter specifies the starting address of the buffer chain to be displayed. The value may be a memory address or an entry point name (externally defined variable) whose address is used as the base address for the display.

DISPLAY_OPTION (DO)

This parameter specifies the amount of data to be displayed. If FULL (or EXPANDED) is specified, the descriptors and data for all messages in the queue are displayed. If BRIEF (or SUMMARY) is specified, only summary information indicating the starting address of the descriptor for each message chain, the number of buffers chained for each message, and the total characters in the message are displayed.

If the DISPLAY_OPTION parameter is not specified, FULL is assumed.

REPEAT_COUNT (RC)

This parameter specifies the number of messages to be displayed. If the keyword ALL is specified, or if REPEAT_COUNT is greater than the number of existing messages, all messages from the specified starting descriptor are displayed (up to a maximum of 4096 messages).

If REPEAT_COUNT is omitted, all message chains from the specified starting descriptor are displayed (up to a maximum of 4096 calls).

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand writes information for the QCB located at address 175242(16) into the file OUTFILE.

```
DISPLAY_DATA_QUEUE  ADDRESS=175242  OUTPUT=OUTFILE
```

or

```
DISDQ A=175242 O=OUTFILE
```

Output from this subcommand is formatted as follows:

```
QUEUE CONTROL BLOCK DISPLAY

QCB ADDRESS                                              175242(16)
  LENGTH       { LENGTH OF QUEUE }                         279(16)
  COUNT        { NUMBER OF ENQUEUES }                     1A1B(16)
  QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }        1A6878(16)
  QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }         1A37C8(16)
  QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }             0(16)


BUFFER CHAIN DISPLAY

DATA DESCRIPTOR      1A6878(16)    NEXT_DESCRIPTOR      1151CE(16)
NEXT_MESSAGE         192539(16)    THE_DATA             122872(16)
OFFSET                  4F(16)     COUNT_BUFFER            42(16)
COUNT_MESSAGE           7F(16)     USAGE_DESCRIPTOR      FFFF(16)

USAGE COUNT            FFFF(16)
DATA AT THE_DATA:

  122872  FFFF  2020  2020  2020  2020  2020  2020  2000
  122882  0000  0020  2020  2000  0000  0020  2020  2020
  122892  2020  2020  2020  2020  2020  2020  2020  2000
  1228A2  0000  0020  2020  2000  0000  0020  2020  2020
  1228B2  2020  6465  666C  2C20  362C  2031  2C20  6173    def1, 6, 1, as
  1228C2 [5010  132F  42C9  0012  0001  0008  3501  0185  P  /B      5
  1228D2  0114  1336  1426  8000  0000  0108  0025  1000    6 &      %
  1228E2  7302  5D54  4449  3131  3520  2020  2020  2020  s ]TDI115
  1228F2  2020  2020  2020  2020  2020  2020  2020  2020
  122902  2020]


DATA DESCRIPTOR      1151CE(16)    NEXT_DESCRIPTOR      18FB2A(16)
NEXT_MESSAGE            0(16)      THE_DATA             1151F4(16)
OFFSET                  1(16)      COUNT_BUFFER            3D(16)
COUNT_MESSAGE           3D(16)     USAGE_DESCRIPTOR      FFFF(16)

USAGE COUNT            FFFF(16)
DATA AT THE_DATA:

  1151F4  FFFF [0B01  02A4  1216  4641  494C  5552  455F          FAILURE_
  115204  4D41  4E41  4745  4D45  4E54  5F42  4153  451A  MANAGEMENT_BASE
  115214  1F00  0000  001A  1F00  0000  001A  1F00  0000
  115224  001A  1F00  0000  000A  0100  010A  0113  02]
            .     .     .     .     .
            .     .     .     .     .
            .     .     .     .     . .
```

The rest of the data for this message is displayed, then all other messages in the queue follow.

The following subcommand displays summary information for the QCB located at address 175242(16).

```
DISDQ  A=175242 DO=B
```

Output from this subcommand is formatted as follows:

```
QUEUE CONTROL BLOCK DISPLAY

QCB ADDRESS                                                  175242(16)
  LENGTH       { LENGTH OF QUEUE }                             279(16)
  COUNT        { NUMBER OF ENQUEUES }                          1A1B(16)
  QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }             1A6878(16)
  QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }              1A37C8(16)
  QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }                  0(16)

          STARTING ADDRESS   NUMBER OF   TOTAL CHARACTERS PER MESSAGE
           OF DESCRIPTOR      BUFFERS          (COUNT_MESSAGE)

MESSAGE # 1    175242(16)         1          2A(16)=     42(10)
MESSAGE # 2    192539(16)        2B9        8E86(16)=  36486(10)
MESSAGE # 3    1A37C8(16)        12B       3547BC(16)= 491772(10)
```

If you want, the DISPLAY_BUFFER_CHAIN command could next be used to display the data in the buffer chains for particular messages by using the addresses given above, or DISDQ could be used with the DO=FULL option to display the data for all the messages.

Message chains that contain zero byte buffers or invalid buffer_count data are flagged with an alert following the message summary line, as in the following example:

```
   ...
MESSAGE # 7    1E00A2(16)          1C        24F(16)=    1103(10)
     ****    ZERO BYTE BUFFER ENCOUNTERED IN PRECEDING MESSAGE
   ...
```

DISPLAY_DI_SYSTEM_STATUS (DISDSS) ANACD Subcommand

Purpose This subcommand displays system configuration information. Its display is similar to that of the command by the same name that can be sent through the Network Operator Utility (NETOU) to active DIs.

NOTE

This subcommand applies to the ICA-II as well as the DI. You may also display ICA-II system configuration information by entering the DISPLAY_ICA_SYSTEM_ STATUS (DISISS) subcommand, an alias for the DISDSS subcommand. DISISS is described in this chapter.

Format

DISPLAY_DI_SYSTEM_STATUS
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand writes DI system status information to file STOUT.

```
DISPLAY_DI_SYSTEM_STATUS OUTPUT=STOUT
```

or

```
DISDSS O=STOUT
```

Output from this subcommand is formatted as follows:

```
DISPLAY DI SYSTEM STATUS

System name       = MDI_8C
System identifier = 08002510008C(16)     Master clock     =        FALSE
Release level     =         2404         Number of tasks  =           57
Boot version      =         2404         CPU utilization  =           11 %
DI loaded from  MCI board in slot 7      Helping system   = 080025C00107(16)

Buffers
              free        total         size
Data          1748         2102          144          State is GOOD
Desc           654          700           32

Memory      (2 MB Configuration)
              free    fragments    deloadable
MPB            540            1             0
PMM           3392            5             0          State is GOOD
SMM         227710           78         69576
RESERVED      1000            1           N/A

50% of memory after configuration made into buffers
Largest SMM memory fragment available =   210040
```

Remarks The following definitions apply to the DISDSS display:

System Name	Configuration file name of the DI (from DEFINE_SYSTEM command).
System Identifier	12-digit hexadecimal system identifier associated with the DI.
Release Level	Level at which the software was compiled.
Boot Version	Software version number specified in the exception list file (on the INITMDI command for NOS MDIs, and DEFES or DEFBD in exception file for other DIs).
Master Clock	Indicates whether this DI maintains the network's master clock.
Number of Tasks	Number of tasks executing in the DI.
CPU Utilization	Percentage of time the CPU is active.
Helping System	DI system that facilitated last reload.
Buffers	*Data.* Buffer used to store user data. A configurable percentage of data buffers is allocated when the DI is initialized. *Descriptor.* Buffer used to keep track of data buffer information, such as chaining, usage counts, and so on. *Free.* Number of allocated buffers not currently in use. *Total.* Total number of buffers that are allocated for use by the DI. *Size.* Size in bytes of the buffer type (set by DEFINE_SYSTEM command).

Memory	*MPB*. System memory that resides on the MPB board. *PMM*. System memory that resides on the PMM board. *SMM*. System memory that resides on the SMM board. *Reserved*. Used by the Executive to communicate with other executive services when there is no other memory available. *Free*. Amount of system main memory that is currently available for modules to be loaded into. *Fragments*. Number of segments of available memory. *Deloadable*. Memory taken up by nonactive tasks that can be used when memory space becomes critically short.
Buffer/Memory State	Four states indicating the relative availability of buffers and memory in the DI: GOOD, FAIR, POOR, and CONGESTED. The boundaries between these states are set during configuration (DEFINE_SYSTEM command). Each boundary is expressed as a percentage of the total resource allocated after DI configuration.

DISPLAY_EXECUTIVE_ERROR_TABLE (DISEET) ANACD Subcommand

Purpose This subcommand displays information from the executive error table.

Format

DISPLAY_EXECUTIVE_ERROR_TABLE
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand writes executive error table information to file EXEET.

```
DISPLAY_EXECUTIVE_ERROR_TABLE OUTPUT=EXEET
```

Display from this subcommand is formatted as follows. Refer to the Remarks following this display for explanations of some of the fields.

```
EXECUTIVE ERROR TABLE DISPLAY

Stopped by module EXEC_PMM + 312 (16), supervisor stack = 3EF0 (16)
SMM error count by slot = [0, 0, 1, 2, 0, 0, 0]
MPB status register shows battery low warning
Number of spurious interrupts = 2

Error # 1:  privilege violation
Milliseconds since error    =      512
OPERATOR_SUPPORT_APPLICATION    +  3E0(16) Task is in user state
Running task identifier     =   102FE0(16) Supervisor stack = 3ED2 (16)

     Registers at time of error

D0-D7 00000000 00000001 00000000 00004A7A 0000FFFF 00000001 0000FFFF 0016A084
A0-A7 001ADB14 00000C76 00000020 00000000 00010000 001AD0AE 001ADA70 001ADA6C

Error # 2:  bus error
Milliseconds since error    =     1498
OPERATOR_SUPPORT_APPLICATION    +  3E0(16) Task is in user state
Running task identifier     =   102FE0(16) Supervisor stack = 3ED2(16)
Instruction register        =    4E20(16) Access address = 102555(16)
MPB status register shows parity error      First failure addr = 102553(16)
Access code                 =  PMM bus      Access type    = Read io

     Registers at time of error

D0-D7 00000000 454C4C45 4E2C0063 616E0059 4F550068 65617200 6D653F00 00000000
A0-A7 001CCB12 00102555 00000000 00000000 00000000 001EF7AE 001CCEF0 001AEAEC

Error # 3:  smm double bit error
Milliseconds since error    =     2560
OPERATOR_SUPPORT_APPLICATION    +  3E0(16) Task is in supervisor state
Running task identifier     =   102FE0(16) User stack =  10ED2 (16)
SMM (slot 2) error log      =    3A4F(16) Slot  7 was accessing SMM

     Registers at time of error

D0-D7 00000000 00000054 656C6C00 6D650069 66004927 6D007361 6E652E00 41524259
A0-A7 001ADB14 00000C76 00000020 00000000 00010000 001AD0AE 001ADA70 001ADA6C
```

Remarks Lines 1 through 4 of the executive error table display are written only if they apply to conditions at the time of the dump.

Line 3 of the display, if written, can indicate any of the following warnings from the MPB status register:

- Battery low
- Temperature high
- Temperature shutdown
- AC low

The first line of information about an error indicates one of the following:

- Bus error
- Address error
- Illegal instruction
- Zero divide
- Check instruction
- Arithmetic overflow
- Privilege violation
- Bad status register
- SMM single-bit error
- SMM double-bit error
- SMM detected bus parity error
- Task runs too long
- XX, an unrecognized code

The third line of information about an error indicates whether the task in error was in user or supervisor state.

Up to four errors are displayed, in chronological order from the most recent to the least recent.

NOTE

The DISPLAY_EXECUTIVE_ERROR_TABLE subcommand displays the errors in chronological order even though the order of the error buffers in the executive error table data structure is not necessarily chronological.

Error #2 in the example on the preceding page shows the format of an additional display line for address or bus errors. This additional line is formatted as follows:

```
Instruction register    =   xxxx(16) Access address = xxxxxxxx(16)
```

For bus errors, the MPB status register is examined and a line is displayed that reports the first failure address and whether the error was dtack_time_out, bus_lock_time_out, or parity_error. If the error was due to parity or dtack_time_out, the access code and type fields from the MPB status register are also extracted and displayed, as in Error #2 from the preceding example.

Error #3 in the preceding example shows the format of an additional display line for SMM single- or double-bit errors. This additional line is formatted as follows:

```
SMM (slot xx) error log      =     xxxx(16) Slot xx was accessing SMM
```

MPB-II Bus Error Recovery Register

The MPB-II bus error recovery register, that may be displayed when a bus error has caused a DI reset, has a 16-bit register with the following title:

```
Bus err recovery reg=xxxx(16)
```

The bits in this entry are shown in the following table:

Bit	Description
Bit 15	Error during access to local MPB-II
Bit 14	Error during access to Internal Transfer Bus
Bit 13	Error during write
Bit 12	Error during interrupt acknowledge cycle
Bit 11	Data acknowledge timeout
Bit 10	Unused
Bit 9	Internal Transfer Bus read parity error
Bit 8	Unused
Bit 7	Unused
Bit 6	Unused
Bit 5	Unused
Bit 4	Bus grant timeout (BGTO)
Bits 3-0	Hexadecimal address of card that was bus master. Valid only during BGTO.

DISPLAY_HARDWARE_STATUS (DISHS) ANACD Subcommand

Purpose This subcommand displays information about the state and status of the hardware installed in the DI when the DI reset. Its display is similar to that of the command by the same name that can be sent through the Network Operator Utility (NETOU) to active DIs.

Format

DISPLAY_HARDWARE_STATUS
DEVICE_NAME = name or *keyword*
DISPLAY_OPTION = keyword
OUTPUT = file_name
STATUS = status_variable_name

Parameters *DEVICE_NAME (DN)*

This parameter specifies the name of the LIM board or the ICA hardware for which information is to be displayed. When specifying a LIM, device names are of the form $nnnnx where nnnn is a 3- or 4-character alphabetic board name, and x is a single-digit board slot number. For example, the LIM in slot 7 would be entered $LIM7. If a LIM is specified with this parameter, then only status for that LIM is displayed.

To display the ICA hardware status, enter $ICA_II to display ICA_II hardware. Enter $ICA_ICI to display the ICA_II channel interface. Enter $ICA_IEI to display the ICA_II Ethernet Interface.

If no DEVICE_NAME is specified, if DN=ALL, or if the specified DEVICE_NAME is not recognized, then the status of all major boards and LIMs is displayed.

DISPLAY_OPTION (DO)

This parameter indicates the extent of the data to be displayed. If the keyword BRIEF (B, SUMMARY, or S) is specified, summary information is displayed for all major boards and only basic summary information on LIM status is displayed. If the keyword FULL (F, EXPANDED, or E) is specified, expanded status information is displayed for the LIMs, in addition to the basic display.

If the DISPLAY_OPTION parameter is not specified, then BRIEF is used.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following examples are produced when displaying the LIM hardware status.

The following subcommand displays a brief summary of DI LIM hardware status:

```
DISPLAY_HARDWARE_STATUS DISPLAY_OPTION=BRIEF
```

or

```
DISHS DO=B
```

Output from this subcommand is formatted as follows:

```
DI HARDWARE STATUS

      Card  Card     Boot                          Version    ROM     Dump
Slot  type   ok  enabled  State      Status          (16)   level  address (16)
-------------------------------------------------------------------------------
  0    MPB  yes       no     on  configured           0     50C        0
  1    PMM  yes       no     on  configured           8     50C    10000
  2    SMM  yes       no     on  configured           2       0   100000
  3    SMM  yes       no     on  configured           0       0   200000
  4    CIM  yes       no     on  configured           0     50C    80000
  5  EMPTY
  6   ESCI  yes       no     on  not config          0     806    A0000
  7    MCI  yes      yes     on       active         0     50C    B0000

Line Interface Modules

Slot            State      Status  Parent CIM  LIM Type
-------------------------------------------------------
  0               on  configured       $CIM4  rs232
  1               on  configured       $CIM4  rs232
  2               on  configured       $CIM4  rs232
  3               on  configured       $CIM4  rs232
  4               on  configured       $CIM4  rs232
  5                                            lim_slot_empty
  6                                            lim_slot_empty
  7               on  not config       $CIM4  rs449
```

If DO=FULL is selected, then, in addition to the above output, expanded LIM status is shown in the following format:

```
Expanded LIM status

LIM  PORT  State      Status  Protocol       Line Name
-------------------------------------------------------
 0      0     on     enabled  ASYNC          LINE00
        1     on     enabled  ASYNC          LINE01
        2     on     enabled  ASYNC          LINE02
        3     on     enabled  ASYNC          LINE03
 1      0     on     enabled  ASYNC          LINE10
        1     on     enabled  ASYNC          LINE11
        .      .        .          .
        .      .        .          .
        .      .        .          .
```

If DEVICE_NAME=$LIM1 and DISPLAY_OPTION=FULL, then the following is displayed:

```
DI HARDWARE STATUS

Line Interface Modules

Slot              State      Status  Parent CIM  LIM Type
---------------------------------------------------------
   1                 on  configured       $CIM4  rs232

Expanded LIM status

LIM  PORT  State      Status  Protocol        Line Name
-------------------------------------------------------
  1     0     on     enabled  ASYNC           LINE10
        1     on     enabled  ASYNC           LINE11
        2     on     enabled  ASYNC           LINE12
        3     on     enabled  ASYNC           LINE13
```

If DN=$LIM1 and DO=BRIEF, then only the following is displayed:

```
DI HARDWARE STATUS

Line Interface Modules

Slot              State      Status  Parent CIM  LIM Type
---------------------------------------------------------
   1                 on  configured       $CIM4  rs232
```

Examples The following examples are produced when displaying the ICA_II hardware status.

The following subcommand displays the default value which is reported for the ICA_II as one board in slot 0.

```
DISHS
```

Output from this subcommand is formatted as follows:

```
      Card  Card   Boot                            Version   ROM     Dump
Slot  type    ok  enabled  State      Status        (16)    level  address (16)
--------------------------------------------------------------------------------
  0   $ICA   yes     n/a      on  configured          0       0       N/A
```

If the DN=$ICA, the following output is displayed:

```
DI HARDWARE STATUS


Device Name   Port    State       Status
----------------------------------------
$ICA            0       on    configured
```

If the DN=$ICA_ICI, the following output is displayed:

```
DI HARDWARE STATUS

ICA-II Channel Interface Status
-------------------------------

Device Name   Port    State        Status
-----------------------------------------
$ICA_ICI        0       on     configured
```

If DN=$ICA_IEI is entered, the following output is displayed:

```
DI HARDWARE STATUS

ICA-II Ethernet Interface Status
--------------------------------

Device Name   Port    State        Status
-----------------------------------------
$ICA_IEI        0       on     configured
```

DISPLAY_ICA_SYSTEM_STATUS (DISISS) ANACD Subcommand

Purpose This subcommand displays system configuration information.

Format

DISPLAY_ICA_SYSTEM_STATUS
OUTPUT = file_name
STATUS = status_variable_name

Remarks This subcommand is an alias for the DISDSS subcommand. DISDSS is described in this chapter.

DISPLAY_LINE_CONTROL_BLOCK (DISLCB) ANACD Subcommand

Purpose This subcommand displays the contents of selected fields of the configured line control block (CLCB) for a given line or all lines.

Format

DISPLAY_LINE_CONTROL_BLOCK
LINE_NAME = line_name or *keyword*
OUTPUT = file_name
STATUS = status_variable_name

Parameters *LINE_NAME (LN)*

This parameter specifies the name of the line (up to 31 characters) for which CLCB information is desired. Line names are assigned in the Configuration File or with interactive NETOU commands (see the CDCNET Configuration Guide). The keyword ALL causes information to be displayed for all configured lines.

If LINE_NAME is omitted, information is displayed for all configured lines.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable.

Examples The following subcommand writes information about the line named LINE01 to output file OUTFILE.

```
DISPLAY_LINE_CONTROL_BLOCK  LINE_NAME=LINE01 OUTPUT=OUTFILE
```

or

```
DISLCB LN=LINE01 O=OUTFILE
```

Output from the DISPLAY_LINE_CONTROL_BLOCK subcommand is formatted as follows:

```
CONFIGURED LINE CONTROL BLOCK     AT ADDRESS      1423DC(16)

   POINTER TO FIRST TCCB                               0(16)
   POINTER TO ACTIVE LCB                           15AA20(16)
   POINTER TO NEXT CLCB                            140FF8(16)
   OPTIONAL TIP EXTENSION POINTER                      0(16)
   LINE NAME                                        LINE01
   LINE INTERFACE MODULE                                1
   PORT NUMBER                                          1
   TIP TYPE                                      ASYNC TIP
   LINE TYPE                                     DEDICATED
   FRAMING TYPE                                      ASYNC
   CARRIER TYPE                                   CONSTANT
   LINE SPEED                                         1200
   ASYNC AUTOREC TYPE                  SPEED AND CODE SET
   CONNECT TIME TIMEOUT                                 0
   DISCONNECT TIMEOUT                                  30
   USER CONNECTION LIMIT                                4
   EIA FLOW CONTROL                                  TRUE
   EFC CLOCKING                                         3
   LCSM TASK ID                                    1421C8(16)
   TIP TASK ID                                         0(16)
   CONFIGURATION CMD QUEUE                             0(16)
   ADD CB COUNT                                         0
   LCM STATE                                            1
   LINE DOWN REASON                  VALUE =           FF(16)
   AUTOREC TIP TYPE                              ASYNC TIP
   AUTOREC LINE SPEED                                 AUTO
   AUTOREC CODE SET                                   AUTO
   AUTOREC PARITY                                     AUTO
   CONNECT TIMER                                        0
   LCSM LINE ENABLE STATUS                           TRUE
   LCSM STATE                                           2
   LCSM AUTOREC STATE                LINE NOT INITIALIZED
```

DISPLAY_LINKED_LIST (DISLL) ANACD Subcommand

Purpose This subcommand displays a list of linked elements beginning at the specified machine address. Use DISPLAY_LINKED_LIST to track a chain of data structures in which the pointer to the next element in the chain resides at a known location within each data structure.

Format

DISPLAY_LINKED_LIST
ADDRESS = address or entry point name
LINK_OFFSET = integer
BYTE_COUNT = integer
REPEAT_COUNT = integer or *ALL*
HEADER = integer
OUTPUT = file_name
STATUS = status_variable_name

Parameters

ADDRESS (A)

This required parameter specifies the starting address of the linked list to be displayed. The value may be a memory address or an entry point name (externally defined variable) whose address is used as the base address for the display. Entry point names are found by using the DISPLAY_MEMORY_MAP subcommand described later in this chapter.

LINK_OFFSET (LO)

This parameter specifies the location within each linked list element of the pointer to the next element. LINK_OFFSET is expressed as the number of bytes from the beginning of each element in the linked list to the pointer to the next element.

If LINK_OFFSET is omitted, the value 0 is assumed.

BYTE_COUNT (BC)

This parameter specifies the number of bytes to display for each element in the linked list.

If BYTE_COUNT is omitted, the value 48 is assumed.

REPEAT_COUNT (RC)

This parameter specifies the number of elements to display from the linked list. The keyword ALL causes all elements in the linked list to be displayed.

If REPEAT_COUNT is omitted, all linked entries are displayed.

HEADER (H)

This parameter specifies the number of bytes to display as a header preceding the starting address of each linked list element.

If HEADER is omitted, a value of 0 is assumed (no header is displayed).

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples

The following subcommand displays three elements of a linked list with 6 bytes of header with each entry. The starting address of the list is 56E.

```
DISPLAY_LINKED_LIST ADDRESS=56E REPEAT_COUNT=3 HEADER=6
```

Output from this subcommand is formatted as follows:

```
DISPLAY_LINKED_LIST

START ADDRESS:      56E

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

     568  1B3E 0016 5CFC                             > \

     56E  0010 B950 0000 0042 0000 0000 0000 0000      P   B
     57E  016C 2700 0000 0090 03E8 67FE 0000 2D82    1'       g   -
     58E  0010 6FD8 4717 0000 0000 0000 0594 016C     o G           1

ADDRESS OF NEXT ELEMENT:   10B950

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

  10B94A  80AE 0000 0000

  10B950  0011 F1CC 0000 0042 0000 0000 434F 4E46          B    CONF
  10B960  4947 5552 4154 494F 4E5F 5354 4154 5553   IGURATION_STATUS
  10B970  5F52 4550 4F52 5445 5220 2000 0003 0232   _REPORTER       2

ADDRESS OF NEXT ELEMENT:   11F1CC

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

  11F1C6  80AE 0000 0000

  11F1CC  0011 954C 0000 0016 0000 0000 4F4E 5F4C      L         ON_L
  11F1DC  494E 455F 4C4F 4144 4552 2020 2020 2020   INE_LOADER
  11F1EC  2020 2020 2020 2020 2020 2000 0003 0232                  2
```

The same linked list without the header displayed would be shown in the following format:

```
DISPLAY_LINKED_LIST

START ADDRESS:      56E

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

     56E  0010 B950 0000 0042 0000 0000 0000 0000      P   B
     57E  016C 2700 0000 0090 03E8 67FE 0000 2D82    1'       g   -
     58E  0010 6FD8 4717 0000 0000 0000 0594 016C     o G           1

ADDRESS OF NEXT ELEMENT:   10B950

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

  10B950  0011 F1CC 0000 0042 0000 0000 434F 4E46          B    CONF
  10B960  4947 5552 4154 494F 4E5F 5354 4154 5553   IGURATION_STATUS
  10B970  5F52 4550 4F52 5445 5220 2000 0003 0232   _REPORTER       2

ADDRESS OF NEXT ELEMENT:   11F1CC

HEX ADDR            HEXADECIMAL DATA                  ASCII DATA

  11F1CC  0011 954C 0000 0016 0000 0000 4F4E 5F4C      L         ON_L
  11F1DC  494E 455F 4C4F 4144 4552 2020 2020 2020   INE_LOADER
  11F1EC  2020 2020 2020 2020 2020 2000 0003 0232                  2
```

DISPLAY_LOG_QUEUES (DISLQ) ANACD Subcommand

Purpose

The DISPLAY_LOG_QUEUES subcommand displays information about available or specified log queues, including the log identifier and variable parts of the messages in the queues. For all types of queues, the messages are displayed in chronological order.

NOTE

The USE_TEMPLATES parameter, an optional parameter that formats the messages according to their templates, is a valid part of the command format but is not functional in this release.

Format

DISPLAY_LOG_QUEUES
QUEUE_TYPE=keyword
DISPLAY_OPTION=keyword
USE_TEMPLATES=boolean
OUTPUT=file_name
STATUS=status_variable_name

Parameters

QUEUE_TYPE (QT)

This parameter specifies the type of queue to display, from the following options:

PRESERVE (P)

The log message preserve queue saves the 15 most recent log messages that have been processed by the Log Support Application. These messages can be displayed from the dump file using this selection for the QT parameter.

If a DI resets before initialization completes, the preserve queue may not yet be active. For the resulting dump file, this selection displays an informative message to that effect.

INITIAL (INITIALIZATION or *I)*

The log message initialization queue maintains log messages during the DI initialization process. If initialization fails to complete, these messages can be displayed from the dump file using this selection for the QT parameter.

If a DI resets after initialization, the initialization queue is not active. For the resulting dump file, this selection displays an informative message to that effect.

ITM

This option is not functional in this release of CDCNET.

ALL

All available log message queues are displayed. If both the preserve and initialization log queues were active when the DI reset, both queues are displayed with this option.

If no queue type is specified, then all the active queues currently supported by this subcommand are displayed.

DISPLAY_OPTION (DO)

This parameter specifies the detail of data to be displayed for the selected queue type. If the keyword BRIEF (B, SUMMARY, or S) is specified, the date, time, system identifier and log identifier are displayed for each message.

If the keyword FULL (F, EXPANDED, or E) is specified, all template identifiers and variable data for each log message are displayed.

If DISPLAY_OPTION is not specified, a full display results.

USE_TEMPLATES (UT)

This parameter is not currently functional, but its position in the DISPLAY_LOG_QUEUES subcommand format is reserved. If you want to use the OUTPUT or STATUS parameters for this subcommand, either specify their names (for example, O=QOUT), or skip over the UT parameter with a comma (for example, DISLQ,P,F,,QOUT).

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks The date, time, log and template identifiers are displayed in decimal. The system address (system identifier) is displayed in hexadecimal. Variable data is displayed as it would be in the log file.

See the CDCNET online Diagnostic Messages manual for descriptions of the log messages found in these queues.

Examples The following subcommand displays the initialization and preserve log queues, if active, in BRIEF format:

```
DISLQ DO=B
```

Output from this subcommand is formatted as follows:

```
PRESERVE LOG QUEUE

  DATE       TIME           SYSTEM ID       LOG ID
  ------------------------------------------------

  87/11/12   14.02.56.071   080025100078      618
  87/11/12   14.07.50.409   080025100078      618
  87/11/12   14.09.35.622   080025100078      618
  87/11/12   14.13.43.929   080025100078      618
  87/11/12   14.14.14.007   080025100078      620
  87/11/12   14.14.15.698   080025100078      618
  87/11/12   14.16.48.683   080025100078      620
  87/11/12   14.17.53.305   080025100078      618
  87/11/12   14.20.17.082   080025100078      620
  87/11/12   14.20.22.493   080025100078      618
  87/11/12   14.23.54.781   080025100078      620
  87/11/12   14.28.38.571   080025100078      619
  87/11/12   14.28.52.266   080025100078      617
  87/11/12   14.28.56.149   080025100078      617
  87/11/12   14.31.03.281   080025100078      620
```

The following subcommand displays the preserve log queue in FULL format:

```
DISLQ QT=P DO=F
```

Output from this subcommand is formatted as follows:

```
PRESERVE LOG QUEUE

  DATE      TIME           SYSTEM ID       LOG ID
  ------------------------------------------------

  87/11/12  14.02.56.071   080025100078      618
    Template ID  =  155
    Variable data fields:
      1     VE102
      2     4
      3     3
      4     L07843
      5     38400
      6     DED
      7
      8     $CONSOLE_100078_4300000000
      9     0
     10     0
     11     CON
     12     ASYNCTIP
     13     ASYNC
     14     0000A002080025100078AB1B
     15     009901020800250000665091

  87/11/12  14.07.50.409   080025100078      618
    Template ID  =  155
    Variable data fields:
      1     S01
      2     5
      3     2
      4     L07852
      5     19200
      6     SW
      7
      8     $CONSOLE_100078_5200000000
      9     0
     10     0
     11     CON
     12     ASYNCTIP
     13     ASYNC
     14     0000A002080025100078AB1B
     15     00000063080025300OD1FEBF

       .         .        .       .
       .         .        .       .
       .         .        .       .
```

DISPLAY_MEMORY (DISM) ANACD Subcommand

Purpose The DISPLAY_MEMORY subcommand displays memory contents beginning at the specified machine address or externally defined entry point. Display is in both hexadecimal and ASCII formats.

Format

DISPLAY_MEMORY
ADDRESS = address or entry point name
BYTE_OFFSET = integer
BYTE_COUNT = integer
REPEAT_COUNT = integer or *keyword*
OUTPUT = file_name
STATUS = status_variable_name

Parameters

ADDRESS (A)

This required parameter specifies the base address used for memory display. It may be specified as either a numeric machine address or an externally defined entry point name. Entry point names are found by using the DISPLAY_MEMORY_MAP subcommand, described in this chapter.

BYTE_OFFSET (BO)

This parameter specifies the offset to the base address established by the address parameter. Its value is added to the base address to form a new address.

If BYTE_OFFSET is omitted, display begins from the base address.

BYTE_COUNT (BC)

This parameter specifies the number of bytes to be displayed, 16 bytes to a line. Or, in conjunction with the REPEAT_COUNT parameter, it specifies the length in bytes of each item to be displayed. Using BYTE_COUNT and REPEAT_COUNT together lets you display memory in groups of bytes, such as when a data structure of known length is repeated through memory.

If BYTE_COUNT is omitted, a value of 1 is assumed.

REPEAT_COUNT (RC)

This parameter specifies the number of items of length BYTE_COUNT to be displayed. The keyword ALL causes all memory from the specified address to the end of memory to be displayed. If no byte count is specified, REPEAT_COUNT specifies the number of bytes to be displayed, 16 bytes to a line.

If REPEAT_COUNT is omitted, only one item of length BYTE_COUNT is displayed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks The REPEAT_COUNT parameter produces the most readable display if only one form of count is required, since an address is shown for every line of output.

If both the REPEAT_COUNT and BYTE_COUNT parameters are omitted, then 128 bytes (eight lines of 16 bytes each) are displayed rather than just one byte. (See the examples.)

Examples The following subcommand displays 48 one-byte long items, starting at hexadecimal address 8400.

```
DISPLAY_MEMORY ADDRESS=8400 RC=48
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:  00008400

HEX ADDR                HEXADECIMAL DATA                ASCII DATA

00008400   0007 0007 005A 0002 0000 0042 0000 0020        Z    B
00008410   0000 0800 0013 B116 0000 0000 FF00 0000
00008420   0040 0000 0042 0000 001A 0010 0000 7FFF     @    B
```

The following subcommand displays five items, where each item is 20 bytes long, starting at decimal address 100.

```
DISM A=100(10) BC=20(10) RC=5
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:  00000064

HEX ADDR                  HEXADECIMAL DATA                  ASCII DATA

00000064   0000 0000 0000 0000 0000 0000 0000 0000
           0000 0000
00000078   0000 0000 0000 0000 0000 0000 0000 0000
           0000 0000
0000008C   0000 0000 0000 0000 0000 0000 0000 0000
           0000 0000
000000A0   0000 0000 0000 0000 0000 0000 0000 0000
           0000 0000
000000B4   0000 0000 0000 0000 0000 0000 0000 0000
           0000 0000
```

The following examples show three ways to display 72(16) bytes of data, starting at memory address 400(16).

```
DISM 400(16) BC=72(10)
```

Output from this subcommand is formatted as follows:

```
DISPLAY_MEMORY a=400(16) bc=72(10)

STARTING ADDRESS:      400

HEX ADDR              HEXADECIMAL DATA                  ASCII DATA

    400  0800 2510 0081 6AA1 0000 5C00 0020 0000     %   j   \
         0000 052C 0001 002C 8032 3204 1613 2921      ,    , 22   )
         0000 0000 0000 0000 0000 0000 0600 B542                   B
         0000 0000 0000 0000 0100 0000 0000 0000
         0000 0000 0000 0000
```

The following example uses both the BYTE_COUNT and REPEAT_COUNT parameters.

```
DISM 400 BC=6 RC=12
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:      400

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

    400  0800 2510 0081                               %
    406  6AA1 0000 5C00                              j   \
    40C  002D 0000 0000                                 -
    412  052C 0001 002C                               ,   ,
    418  8032 3204 1613                               2 2
    41E  2921 0000 0000                               )
    424  0000 0000 0000
    42A  0000 0600 B542                                   B
    430  0000 0000 0000
    436  0000 0100 0000
    43C  0000 0000 0000
    442  0000 0000 0000
```

The following example uses only the REPEAT_COUNT parameter.

```
DISM 400(16) RC=72(10)
```

Output from this subcommand is formatted as follows:

```
DISPLAY_MEMORY a=400(16) rc=72(10)

STARTING ADDRESS:      400

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

    400  0800 2510 0081 6AA1 0000 5C00 0020 0000    %   j   \
    410  0000 052C 0001 002C 8032 3204 1613 2921     ,   ,  2 2   )
    420  0000 0000 0000 0000 0000 0000 0600 B542                   B
    430  0000 0000 0000 0000 0100 0000 0000 0000
    440  0000 0000 0000 0000
```

The following subcommand uses neither the REPEAT_COUNT nor the BYTE_COUNT parameters:

```
DISM 400
```

Output from this subcommand is formatted as follows:

```
STARTING ADDRESS:      400

HEX ADDR              HEXADECIMAL DATA                ASCII DATA

    400  0800 2510 0081 6AA1 0000 5C00 0020 0000    %   j   \
    410  0000 052C 0001 002C 8032 3204 1613 2921     ,   ,  2 2   )
    420  0000 0000 0000 0000 0000 0000 0600 B542                   B
    430  0000 0000 0000 0000 0100 0000 0000 0000
    440  0000 0000 0000 0000 0000 0000 0000 0000
    450  0000 0000 0000 0000 0000 0000 0000 0000
    460  0000 0000 0000 0000 0000 0000 0000 0000
    470  0000 0000 0000 0000 0000 0000 0000 0000
```

DISPLAY_MEMORY_HEADER (DISMH) ANACD Subcommand

Purpose This subcommand displays information from the allocation header of a DI memory extent. Included in the display are the type of extent, the extent size, and the owner identifier.

Format

```
DISPLAY_MEMORY_HEADER
    ADDRESS = address
    BEGINNING_OF_EXTENT = keyword
    OUTPUT = file_name
    STATUS = status_variable_name
```

Parameters

ADDRESS (A)

This required parameter specifies an address from a DI memory extent. The allocation header of the extent is displayed. This address need not specify the beginning of the memory extent (see the BEGINNING_OF_EXTENT parameter, described next).

BEGINNING_OF_EXTENT (BOE)

If the ADDRESS you specify is the beginning of a memory extent and this parameter is set to TRUE (ON or YES), response-time for this subcommand is improved by eliminating a search for the beginning of extent.

If this parameter is not specified, or if it is set to FALSE (OFF or NO), then the subcommand processor searches for the allocation header that describes the memory extent containing the address specified by the ADDRESS parameter.

NOTE

The beginning of extent is the start of the usable area in the memory extent. Be careful if you specify BOE=YES that the address you specify is actually the start of the usable area and *not* the start of the allocation header that is located six bytes before it. If you are uncertain, leave off the BOE parameter and let the subcommand processor make the search.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand displays information from the memory extent that starts at address 102DAE(16):

```
DISPLAY_MEMORY_HEADER ADDRESS=102DAE BEGINNING_OF_EXTENT=YES
```

Output from this subcommand is formatted as follows:

```
MEMORY HEADER DISPLAY
---------------------

For address:   102DAE(16), Beginning of extent:    102DAE(16)

Allocated              =   yes
Owner identifier       =       F011(16)
Byte size              =         32
Task identifier        =     104970(16)
Description            =   ON_LINE_LOADER
Type                   =   descriptor buffer
```

The following subcommand displays information from the header of the memory extent that contains address 102DB4(16):

```
DISMH A=102DB4
```

Output from this subcommand is formatted as follows:

```
MEMORY HEADER DISPLAY
---------------------

For address:   102DB4(16), Beginning of extent:    102DAE(16)

Allocated              =   yes
Owner identifier       =       F011(16)
Byte size              =         32
Task identifier        =     104970(16)
Description            =   ON_LINE_LOADER
Type                   =   descriptor buffer
```

DISPLAY_MEMORY_MAP (DISMM) ANACD Subcommand

Purpose This subcommand displays the names and locations of the software modules that were in the DI when it was reset, including deloadable modules. For each module, the symbolic name of the module and the date and time the module was created (compiled or assembled), are always displayed. Use the DISPLAY_OPTION parameter to select additional information for display.

Format

DISPLAY_MEMORY_MAP
MODULE = module name or *keyword*
DISPLAY_OPTION = keyword
OUTPUT = file_name
STATUS = status_variable_name

Parameters

MODULE (M)

This parameter specifies the name of a particular module for which information is to be displayed. If the keyword ALL is specified, information is displayed for all modules.

If MODULE is omitted, information is displayed for all modules.

DISPLAY_OPTION (DO)

This parameter specifies the type of information to be included in the memory map. The options are:

DO = ENTRY (E)

Selects display of the following entry-point information:

- Symbolic name of entry-point
- Hexadecimal address of entry-point
- Relative address (offset) of entry-point

DO = SECTION (S)

Selects display of the following section information:

- Section ordinal
- Hexadecimal address of section
- Section length
- Section type

DO = FULL (F)

Selects display of both entry-point and section information (options E and S).

If DISPLAY_OPTION is omitted, DO=E is assumed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks Modules on the deloadable module list are displayed by the DISPLAY_MEMORY_MAP subcommand. In the display, the deloadable module list appears after the loaded module list.

Examples The following subcommand displays module section information for the dump file being analyzed:

```
DISPLAY_MEMORY_MAP DISPLAY_OPTION=SECTION
```

or

```
DISMM DO=S
```

Output from this subcommand is formatted as follows:

```
LOADED MODULE LIST                HEX       HEX
MODULE NAME / SECTION ORDINAL     ADDRESS   LENGTH   SECTION KIND

EXEC_MPB                          February 27, 1985   11:36 PM
  SECTION    0(16)                    1340      1CBC CODE_SECTION
  SECTION    1(16)                    3002       1FE WORKING_STORAGE_SECTION

DELOADABLE MODULE LIST            HEX       HEX
MODULE NAME / SECTION ORDINAL     ADDRESS   LENGTH   SECTION KIND

DISPLAY_DI_SYSTEM_STATUS          1985/02/27          17:45:42
  SECTION    0(16)                  199632       8E2 CODE_SECTION
  SECTION    1(16)                  107D64        46 WORKING_STORAGE_SECTION
  SECTION    2(16)                  190908        10 WORKING_STORAGE_SECTION
  SECTION    3(16)                  19092E        18 WORKING_STORAGE_SECTION
  SECTION    4(16)                  195A86        28 WORKING_STORAGE_SECTION
  SECTION    5(16)                  195D58        CB WORKING_STORAGE_SECTION
```

If you omit the DISPLAY_OPTION parameter (or select DO=E), the DISPLAY_MEMORY_MAP subcommand generates a display in the following format:

```
LOADED MODULE LIST                  HEX          HEX
MODULE/ENTRY_POINT NAME             ADDRESS    OFFSET

TRMSCP                            1985/02/27              22:54:51
  CMD_DISPLAY_TRANSPORT_STATUS       116394  SEC 0(16)    +68A(16)
  CMD_DISTS                          116D32  SEC 0(16)   +1028(16)


DELOADABLE MODULE LIST              HEX          HEX
MODULE/ENTRY_POINT NAME             ADDRESS    OFFSET

DISPLAY_DI_SYSTEM_STATUS          1985/02/27              17:45:42
  CMD_DISPLAY_DI_SYSTEM_STATUS_TA   19090C  SEC 2(16)     +4(16)
  CMD_DISDSS_TA                     190912  SEC 2(16)     +A(16)
  CMD_DISPLAY_DI_SYSTEM_STATUS      199632  SEC 0(16)     +0(16)
  CMD_DISDSS                        1997BE  SEC 0(16)   +18C(16)
```

If you specify DO=F, you get both entry-point and section information.

DISPLAY_MEMORY_USERS (DISMU) ANACD Subcommand

Purpose

This subcommand provides a means of determining the memory usage within the DI as a whole, or by a specific task, owner identifier, or owner description.

An optional parameter, ANACD_DIRECTIVES, gives you the choice of building a Dump Analyzer input file to help you with further dump analysis. When the input file is processed, the contents of owned memory extents are displayed.

Format

DISPLAY_MEMORY_USERS
MEMORY_USER = address, task_name, owner_description or *ALL*
EXTENT_TYPE = keyword
DISPLAY_OPTION = keyword
OUTPUT = file_name
ANACD_DIRECTIVES = file_name
STATUS = status_variable_name

Parameters

MEMORY_USER (MU)

This parameter specifies the identity of the user whose memory characteristics are displayed.

- If an integer in the range 1..0ffff(16) is specified, it is assumed to be an owner_id.[1] Larger integers are interpreted as Task Control Block (TCB) addresses.
- If a name is specified, then it is assumed to be a task name. Data for all owner_ids for TCBs scheduled for the specified task are displayed.
- If the value is a string, it is assumed to be an owner_description and the list of owner descriptions is scanned to get the owner_id.
- If ALL is specified, the memory characteristics of all users are displayed.

If no value is specified, then ALL is assumed.

1. Owner_ids are generated specifically for tracking memory ownership, and are associated with each Task Control Block (TCB); the FULL display option of the DISPLAY_TASK_CONTROL_BLOCK subcommand displays the owner_id for each TCB displayed.

EXTENT_TYPE (ET)

This parameter specifies the type of memory extent whose user(s) is displayed. Keywords are:

MEMORY (M)

Ownership of allocatable memory extents is displayed.

BUFFER (B)

Ownership of buffers is displayed.

ALL

Ownership of both allocatable memory extents and buffers is displayed.

If this parameter is omitted, then ALL is assumed.

DISPLAY_OPTION (DO)

This parameter specifies the amount of data to be displayed. If FULL (or EXPANDED) is specified, information about all types of memory owned by the specified user or all users is displayed. If BRIEF (or SUMMARY) is specified, one line of summary information about buffer usage and one line about memory usage is displayed per user.

If the DO parameter is not specified, then BRIEF is assumed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

ANACD_DIRECTIVES (AD)

This parameter specifies the file to be used later as a Dump Analyzer input file to display the memory extents specified by the MU and ET parameters. When you use this parameter, the Dump Analyzer locates memory extents in accordance with the MU and ET parameters, and then generates directives to display the contents of these memory extents. The file named on this parameter can be edited before it is submitted to the Dump Analyzer as an input file. (See Remarks for further explanation.)

This parameter is only functional when the DISPLAY_OPTION is specified as FULL. If you attempt to use the AD parameter in conjunction with DO=BRIEF, a warning message is issued and the AD parameter is ignored.

If the ANACD_DIRECTIVES parameter is not specified, no directives file is produced.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks If you want a full display of all memory users, this subcommand takes some time to process since the entire dump is searched for the memory that belongs to each owner. The display of memory use for each memory owner is output as it is determined, with pauses before each new owner. To warn interactive users, a message is output at the start of the subcommand. It reads as follows:

```
--WARNING DA 124--  This command may take some time.  There are
[text] owner_ids for which memory use is displayed.
```

If you use the ANACD_DIRECTIVES parameter, take note of the following:

- DISPLAY_BUFFER_CHAIN directives are generated for extents that are determined to be heads of buffer chains. Other buffer types do not generate directives, as they should be included when the whole buffer chain is displayed. The DISBC directives specify DO=F and an output file name of ZZZDABO. The owner identifier and description are appended as comments in the directives file.
- DISPLAY_MEMORY directives are generated for all memory extents, using the extent size as the RC parameter and an output file name of ZZZDAMO. The owner identifier and extent type are appended as comments in the directives file.
- Directives files are suitable for reprocessing by the Dump Analyzer without modification. To submit the directives:
  - Under NOS/VE, use INCLUDE_FILE.
  - Under NOS, start a separate Dump Analyzer session and specify the directives file for the I parameter.

  If you want to edit a directives file to remove unwanted subcommands or to change parameter specifications before submitting the file for processing, use a NOS or NOS/VE file editor. Under NOS/VE, you can use EDIT_FILE to edit the directives file without leaving the current Dump Analyzer session. Under NOS, you must end your current Dump Analyzer session before using a file editor.

Examples The following subcommand displays information, in the BRIEF format, about System Ancestor memory usage:

```
DISPLAY_MEMORY_USERS  MEMORY_USER=SYSTEM_ANCESTOR  DISPLAY_OPTION=BRI-
EF
```

Output from this subcommand is formatted as follows:

```
DISPLAY MEMORY USERS

 MEMORY USAGE

owner  description                  # of extents  total bytes TCB address
-------------------------------------------------------------------------
F000  SYSTEM_ANCESTOR                       111       126614     104120

 BUFFER USAGE

owner  description                  # of buffers  total bytes TCB address
-------------------------------------------------------------------------
F000  SYSTEM_ANCESTOR                         2           64     104120
```

NOTE

If you select the BRIEF display for all memory users, you might see some memory owners described as *** UNKNOWN ***. This occurs when a task terminates without freeing its memory extents, since the owner_id does not correspond with any of the owner_ids found on the active TCB chain.

This situation is not necessarily a problem, as some tasks pass a memory extent pointer to a secondary user before terminating. Loss of memory should be suspected, however, when the concurrent memory state is found to be congested.

The following subcommand displays full memory-use information for all memory users:

```
DISMU MU=ALL DO=F
```

Output from this subcommand is formatted as follows:

```
DISPLAY MEMORY USERS

Owner Identifier      =    1(16)  INTERRUPT LEVEL REQUEST
TCB Address           =   N/A

   address      bytes   type
  ------------------------------------------------------------
    105ED0        70   intertask message from global memory

Owner Identifier      =   21(16)  MEMBER OF LOG MSG QUEUE
TCB Address           =   N/A

   address      bytes   type
  ------------------------------------------------------------
    1023E2        32   head of buffer chain (descriptor)
    107DD6        32   descriptor buffer from global memory
    127270       146   data buffer
    1273EC        32   descriptor buffer
```

If you specify the ANACD_DIRECTIVES parameter on this subcommand, an input directives file is created for use with the ANACD command. The file contains DISPLAY_MEMORY and DISPLAY_BUFFER_CHAIN subcommands, and notes the associated memory owner. It is formatted as follows:

```
DISM A=0105ED0 RC=0070(10) O=ZZZDAMO.$EOI "OWNER:    1 TYPE: ITM/GLOBAL MEM"
DISBC A=01023E2 DO=F O=ZZZDABO.$EOI "OWNER:   21/ MEMBER OF LOG MSG QUEUE"
DISM A=0010006 RC=0032(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=001002C RC=5162(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=0011F0C RC=0906(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=00158CA RC=0108(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=0015AB8 RC=0032(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=0016224 RC=0298(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
DISM A=00170FC RC=1780(10) O=ZZZDAMO.$EOI "OWNER:   50 TYPE: USER MEMORY"
```

The output file named on these directives is ZZZDAMO. Note that directives generated under the NOS operating system do not specify the $EOI file position, since file position is not a component of NOS file names.

DISPLAY_NETWORK_STATUS (DISNS) ANACD Subcommand

Purpose This subcommand displays status information concerning all networks connected to this DI when the DI was reset.

Format

DISPLAY_NETWORK_STATUS
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand displays the network status of the DI at the time of reset:

```
DISPLAY_NETWORK_STATUS
```

Output from this subcommand is formatted as follows:

```
DISPLAY NETWORK STATUS

network_name       = $NET_1
network_type       = Ethernet
network_identifier = 000000001(16)
network_status     = active
network_cost       = 0000A (16)

network_name       = $NET_2
network_type       = X.25
network_identifier = 00000064(16)
network_status     = active
network_cost       = 06FA (16)
```

If no networks were connected to the DI when the dump file was written, then the following message is displayed:

```
No networks defined.
```

Remarks The following values may be seen in the DISPLAY_NETWORK_STATUS display:

network_type	Ethernet, HDLC, X.25, for example
network_status	Active, enabled, congested, configured

NOTE

The value displayed for network_cost is the configured cost of the network, not the dynamically assigned adjusted_cost.

DISPLAY_SYSTEM_CONFIG_TABLE (DISSCT) ANACD Subcommand

Purpose This subcommand displays information from the system configuration table.

Format

DISPLAY_SYSTEM_CONFIG_TABLE
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local. If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks In the displays that follow, QNEXT is now described as "ADDRESS OF FIRST ENTRY IN QUEUE." In previous releases, this read, "ADDRESS OF NEXT ENTRY IN QUEUE."

Examples The following subcommand writes system configuration table information to the file OUTFILE.

```
DISPLAY_SYSTEM_CONFIG_TABLE  OUTPUT=outfile
```

Output from this subcommand is formatted as follows:

```
SYSTEM CONFIGURATION TABLE DISPLAY

DATA BUFFER AVAILABLE COUNT                          2868
DESCRIPTOR BUFFER AVAILABLE COUNT                    1006
DATA SPACE LENGTH  (BYTES)                            144
DESCRIPTOR BUFFER LENGTH  (BYTES)                      32
STANDARD STACK ALLOCATION  (BYTES)                   2048
RUNNING TASK POINTER                              1883D8(16)
FREE GLOBAL MEMORY  (BYTES)                          4660
FREE PRIVATE MEMORY  (BYTES)                        16150
FREE MPB RAM MEMORY  (BYTES)                         2336
FREE GLOBAL MEMORY EXTENTS  (BYTES)                   100
FREE PRIVATE MEMORY EXTENTS  (BYTES)                   11
FREE MPB RAM MEMORY EXTENTS  (BYTES)                    1
DELOADABLE MPB RAM  (BYTES)                             0
DELOADABLE PMM  (BYTES)                                 0
DELOADABLE SMM  (BYTES)                            201506
DELOAD THRESHOLD - MPB RAM  (BYTES)                     0
DELOAD THRESHOLD - PMM      (BYTES)                  4000
DELOAD THRESHOLD - SMM      (BYTES)                 30000
BINARY TIME OF DAY                                1752400
BCD DATE AND TIME                 86/02/01 23.20.34.070
ASSUMED YEAR                                           86
INTERRUPT FIREWALL CHAIN ADDRESS                     0(16)
NUMBER OF DATA BUFFERS                               3247
NUMBER OF DESCRIPTOR BUFFERS                         1082
STACK OVERFLOW AREA ALLOCATION  (BYTES)               240
OVERFLOWING TASK POINTER                             0(16)
PC CHK INSTRUCTION EXECUTION ADDRESS                 0(16)
USP CHK INSTRUCTION EXECUTION ADDRESS                0(16)
LARGEST ADDRESS IN RESERVED MEMORY                18E70(16)
RESERVE RAM MEMORY  (BYTES)                             0
RESERVE GLOBAL MEMORY EXTENTS                           0
MEMORY STATE                                  memory_good
BUFFER STATE                                  buffer_good
```

```
SYSTEM CONFIGURATION TABLE DISPLAY (Continued)

  MPB TYPE                                              MPB
  PMM ALLOCATED ANY            (BYTES)                    0
  AVAILABLE TO ANY             (BYTES)                63536
  CURRENT PMM REQUIRED         (BYTES)                63536
  MAXIMUM PMM REQUIRED         (BYTES)                63536
  DELOAD PMM ANY               (BYTES)                    0

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 0
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                   13A(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 1
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                     0(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 2
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                     0(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0  6)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 3
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                  2676(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 4
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                     0(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 5
    LENGTH        { LENGTH OF QUEUE }                        1(16)
    COUNT         { NUMBER OF ENQUEUES }                  59D6(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }    23611A(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }     23611A(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 6
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                     0(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  SCHEDULED TASKS PRIORITY READY LISTS                PRIORITY = 7
    LENGTH        { LENGTH OF QUEUE }                        0(16)
    COUNT         { NUMBER OF ENQUEUES }                     0(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }        0(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }         0(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)

  GLOBAL MEMORY EXTENT LIST
    LENGTH        { LENGTH OF QUEUE }                       64(16)
    COUNT         { NUMBER OF ENQUEUES }                  E36A(16)
    QNEXT         { ADDRESS OF FIRST ENTRY IN QUEUE }    116C46(16)
    QLAST         { ADDRESS OF LAST ENTRY IN QUEUE }     27FE32(16)
    QCHARACTERS   { NUMBER OF CHARACTERS IN QUEUE }          0(16)
```

```
SYSTEM CONFIGURATION TABLE DISPLAY (Continued)

  PRIVATE MEMORY EXTENT LIST
    LENGTH       { LENGTH OF QUEUE }                          B(16)
    COUNT        { NUMBER OF ENQUEUES }                     130E(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }         10C7A 6)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }         2B762(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

  MPB RAM MEMORY EXTENT LIST
    LENGTH       { LENGTH OF QUEUE }                          1(16)
    COUNT        { NUMBER OF ENQUEUES }                       1(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }        592(16)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }         592(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

  DEFINED INTERRUPTS LIST
    LENGTH       { LENGTH OF QUEUE }                         11(16)
    COUNT        { NUMBER OF ENQUEUES }                      11(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }       1899D0(16)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }        189DD2(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

  DATA BUFFER QUEUE
    LENGTH       { LENGTH OF QUEUE }                        B34(16)
    COUNT        { NUMBER OF ENQUEUES }                     B53F(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }       19EC1A(16)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }        2427B6(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

  DESCRIPTOR BUFFER QUEUE
    LENGTH       { LENGTH OF QUEUE }                        3EE(16)
    COUNT        { NUMBER OF ENQUEUES }                     36DE(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }       25718A(16)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }        2571B0(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

  RESERVE RAM MEMORY EXTENT LIST
    LENGTH       { LENGTH OF QUEUE }                          0(16)
    COUNT        { NUMBER OF ENQUEUES }                       0(16)
    QNEXT        { ADDRESS OF FIRST ENTRY IN QUEUE }          0(16)
    QLAST        { ADDRESS OF LAST ENTRY IN QUEUE }           0(16)
    QCHARACTERS  { NUMBER OF CHARACTERS IN QUEUE }            0(16)

   If a value in the above display is too large to be displayed, a
   string of asterisks is inserted into the value portion of the
   display to indicate this condition.
```

DISPLAY_TASK_CONTROL_BLOCK (DISTCB) ANACD Subcommand

Purpose This subcommand displays the contents of selected fields for one or more active Task Control Blocks (TCBs) in the dump file.

Format

DISPLAY_TASK_CONTROL_BLOCK
TASK_IDENTIFIER = address, task name, or *keyword*
DISPLAY_OPTION = keyword
OUTPUT = file_name
STATUS = status_variable_name

Parameters

TASK_IDENTIFIER (TI)

This parameter specifies the TCBs to be examined. If TASK_IDENTIFIER is an address, information from the TCB at that address is displayed. If TASK_IDENTIFIER is a task name, all TCBs with that task name are displayed. The keyword ALL causes all tasks to be processed, beginning with the system ancestor task.

If TASK_IDENTIFIER is omitted, only the currently selected task is processed. The currently selected task is the task specified by the SELECT_TASK subcommand (described in this chapter). If SELECT_TASK has not been used to specify a task, the task running when the DI was reset is assumed. If no task was running when the DI was reset, the Dump Analyzer issues diagnostic message number 37.

DISPLAY_OPTION (DO)

This parameter specifies the amount of data to be displayed. If BRIEF (or SUMMARY) is specified, summary information is displayed for each TCB. If FULL (or EXPANDED) is specified, a more extensive display is generated for each TCB.

If DISPLAY_OPTION is not specified, BRIEF is assumed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is not specified, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand displays expanded information from all of the TCBs with task name XEROX_INTERNET_ROUTINES.

```
DISPLAY_TASK_CONTROL_BLOCK  TASK_IDENTIFIER=XEROX_INTERNET_ROUTINES DISPLAY_OPTION=FULL
```

Output from this subcommand is formatted as follows:

```
TASK CONTROL BLOCK DISPLAY

TCB ADDRESS                                                    1C4C02(16)
TASK NAME     XEROX_INTERNET_ROUTINES
NEXT_TASK   { CHAIN TO NEXT TASK POINTER }                         0(16)
ID          { TCB IDENTIFICATION }                                 !TCB
STSIZ       { SIZE OF CURRENT STACK SEGMENT }                    8F0(16)
CHLDQ       { TASK POINTER OF NEXT SIBLING }                   1C4B1E(16)
ADULT       { TASK POINTER OF PARENT }                         103B56(16)
CHILD       { TASK POINTER OF CHILD }                              0(16)
STACK       { ADDRESS OF CURRENT STACK SEGMENT }               1CEB94(16)
STATE       { CURRENT STATE }                                      WAIT
PRIORITY    { ACTUAL PRIORITY }                                       1
EXPRESS ITM { NUMBER EXPRESS INTER_TASK MSGS QUEUED }              0(10)
NORMAL ITM  { NUMBER NORMAL INTER_TASK MSGS QUEUED }               0(10)
SR          { STATUS REGISTER }
              TRACE MODE                NO
              M68000 MODE               USER
              INTERRUPT MASK            000
              RESULT EXTENDED           NO
              RESULT NEGATIVE           NO
              RESULT ZERO               NO
              OVERFLOW                  NO
              CARRY                     NO
PC          { PROGRAM COUNTER }                                10A774(16)
A6          { STACK POINTER }                                  1CF410(16)
A7          { STACK POINTER }                                  1CF410(16)
              MODULE NAME               EXEC_PMM

              OFFSET IN CODE SECTION          142(16)
OWNER_ID    { ID GENERATED FOR MEMORY OWNERSHIP }                F01F(16)


TASK CONTROL BLOCK DISPLAY

TCB ADDRESS                                                    1C4B1E(16)
TASK NAME     XEROX_INTERNET_ROUTINES
NEXT_TASK   { CHAIN TO NEXT TASK POINTER }                         0(16)
ID          { TCB IDENTIFICATION }                                 !TCB
STSIZ       { SIZE OF CURRENT STACK SEGMENT }                    8F0(16)
CHLDQ       { TASK POINTER OF NEXT SIBLING }                   1C4A3A(16)
ADULT       { TASK POINTER OF PARENT }                         103B56(16)
CHILD       { TASK POINTER OF CHILD }                              0(16)
STACK       { ADDRESS OF CURRENT STACK SEGMENT }               1CE286(16)
STATE       { CURRENT STATE }                    WAIT FOR ANY MESSAGE
PRIORITY    { ACTUAL PRIORITY }                                       1
EXPRESS ITM { NUMBER EXPRESS INTER_TASK MSGS QUEUED }              0(10)
NORMAL ITM  { NUMBER NORMAL INTER_TASK MSGS QUEUED }               0(10)
SR          { STATUS REGISTER }
              TRACE MODE                NO
              M68000 MODE               USER
              INTERRUPT MASK            000
              RESULT EXTENDED           NO
              RESULT NEGATIVE           NO
              RESULT ZERO               NO
              OVERFLOW                  NO
              CARRY                     NO
PC          { PROGRAM COUNTER }                                10A774(16)
A6          { STACK POINTER }                                  1CEB2E(16)
A7          { STACK POINTER }                                  1CEB2E(16)
              MODULE NAME               EXEC_PMM
              OFFSET IN CODE SECTION          142(16)
OWNER_ID    { ID GENERATED FOR MEMORY OWNERSHIP }                F01E(16)
```

```
TASK CONTROL BLOCK DISPLAY

TCB ADDRESS                                                    1C4A3A(16)
TASK NAME      XEROX_INTERNET_ROUTINES
NEXT_TASK    { CHAIN TO NEXT TASK POINTER }                        0(16)
ID           { TCB IDENTIFICATION }                                !TCB
STSIZ        { SIZE OF CURRENT STACK SEGMENT }                   8F0(16)
CHLDQ        { TASK POINTER OF NEXT SIBLING }                  1C4956(16)
ADULT        { TASK POINTER OF PARENT }                        103B56(16)
CHILD        { TASK POINTER OF CHILD }                             0(16)
STACK        { ADDRESS OF CURRENT STACK SEGMENT }              1CD978(16)
STATE        { CURRENT STATE }              WAIT FOR ANY MESSAGE
PRIORITY     { ACTUAL PRIORITY }                                       1
EXPRESS ITM  { NUMBER EXPRESS INTER_TASK MSGS QUEUED }             0(10)
NORMAL ITM   { NUMBER NORMAL INTER_TASK MSGS QUEUED }              0(10)
SR           { STATUS REGISTER }
               TRACE MODE                NO
               M68000 MODE               USER
               INTERRUPT MASK            000
               RESULT EXTENDED           NO
               RESULT NEGATIVE           NO
               RESULT ZERO               NO
               OVERFLOW                  NO
               CARRY                     NO
PC           { PROGRAM COUNTER }                               10A774(16)
A6           { STACK POINTER }                                 1CE21C(16)
A7           { STACK POINTER }                                 1CE21C(16)
               MODULE NAME               EXEC_PMM
               OFFSET IN CODE SECTION        142(16)
OWNER_ID     { ID GENERATED FOR MEMORY OWNERSHIP }               F01D(16)
```

The following subcommand displays summary information for all TCBs:

```
DISTCB TI=ALL DO=BRIEF
```

Output from this subcommand is formatted as follows:

```
TASK CONTROL BLOCK DISPLAY

TASK NAME                    | TCB ADDRESS |   A6   | PRIORITY | STATE
-----------------------------|-------------|--------|----------|-----------
SYSTEM_ANCESTOR              |      103B56 | 104422 |        7 | WAIT FOR MSG
INITIALIZATION_M_E           |      1D27B2 | 27E056 |        3 | WAIT FOR MSG
INDEPENDENT_LOG_ME           |      20CA8C | 211C54 |        4 | WAIT FOR MSG
IVT_GATEWAY                  |      20D2DC | 20E35E |        5 | WAIT FOR MSG
TERMINAL_SUPPORT_BOUND       |      200EA0 | 2018FA |        5 | WAIT FOR MSG
LINE_CONTROL_BOUND           |      2004D4 | 200DE8 |        3 | WAIT FOR MSG
NP_GATEWAY                   |      1E7FEC | 1E90A8 |        5 | WAIT FOR MSG
K_DISPLAY_SUPERVISOR         |      1DE294 | 1DE7DC |        3 | WAIT FOR MSG
OPERATOR_SUPPORT_APPLICATION |      1DD5AA | 1DE1EA |        3 | WAIT FOR MSG
INDEPENDENT_FILE_ACCESS_ME   |      1D297A | 1D32AE |        3 | WAIT FOR MSG
MCI_SSR                      |      1C4E16 | 1D0426 |        5 | WAIT FOR MSG
SERVICE_MODULE               |      106E1A | 10774E |        5 | WAIT FOR MSG
XEROX_INTERNET_ROUTINES      |      1C4C02 | 1CF410 |        1 |      WAIT
```

.
.
.

NOTE

All numeric values in the BRIEF display are in hexadecimal.

DISPLAY_TREE (DIST) ANACD Subcommand

Purpose The DISPLAY_TREE subcommand displays a list of linked elements from a binary tree structure, beginning at the specified root address. Data is displayed node-by-node. A tree may also be searched for a specified key, using the SEARCH_KEY parameter.

Format

DISPLAY_TREE
ADDRESS = address or **entry point name**
SEARCH_KEY = integer, name, or *string*
OUTPUT = file_name
STATUS = status_variable_name

Parameters

ADDRESS (A)

This required parameter specifies the root address of the binary tree structure. The value may be a DI memory address or an externally defined entry point name. Entry point names can be found by using the DISPLAY_MEMORY_MAP subcommand described in this chapter.

SEARCH_KEY (SK)

This parameter specifies the node key to be searched for. If a match is found, only the data for that node is displayed; otherwise, a message is issued indicating that no match was found. If SEARCH_KEY is specified, its type must match the tree type.

If SEARCH_KEY is omitted, no key search is performed, and the entire tree is displayed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is not specified, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks It is assumed that the tree being searched is sorted in key order (left link low) and contains no duplicate keys. No checks are made on these conditions during tree display.

Examples The DISPLAY_TREE subcommand shown below displays a loader entry point tree structure with a root address of 108524(16).

```
DIST 108524 O=TREEOUT
```

Output from this subcommand is formatted as follows:

```
Tree Identifier         =   eptb
Number of Nodes in Tree =    809
Tree Kind               = string

Node    1 of  809 (   73 bytes):  key = ABORTASK_

    109F72   4142 4F52 5441 534B 5F20 2020 2020 2020  ABORTASK_
    109F82   2020 2020 2020 2020 2020 2020 2020 20D6
    109F92   0010 AD12 0010 AC54 0000 002C 0000 0000         T     ,
    109FA2   0010 9EAC 0000 0000 0000 0000 0000 0004
    109FB2   0072 0050 0010 A178 00                   r P    x

Node    2 of  809 (   73 bytes):  key = ABORT_MESSAGE

    109F72   4142 4F52 5441 534B 5F20 2020 2020 2020  ABORT_MESSAGE
    109F82   2020 2020 2020 2020 2020 2020 2020 20D6                 O
    109F92   0010 AD12 0010 AC54 0000 002C 0000 0000             B
    109FA2   0010 9EAC 0000 0000 0000 0000 0000 0004       < z 3{
    109FB2   0072 0050 0010 A178 00                   TB 'x8 x

Node    3 of  809 (   73 bytes):  key = ABORT_SYSTEM

    109F72   4142 4F52 5441 534B 5F20 2020 2020 2020  ABORT_SYSTEM
    109F82   2020 2020 2020 2020 2020 2020 2020 20D6                 V
    109F92   0010 AD12 0010 AC54 0000 002C 0000 0000             B
    109FA2   0010 9EAC 0000 0000 0000 0000 0000 0004       R8 w  -
    109FB2   0072 0050 0010 A178 00
```

NOTE

If the structure you attempt to display is not a tree structure, the Dump Analyzer responds with error message number 136.

HELP (HEL)
ANACD Subcommand

Purpose The HELP subcommand displays either a list of all Dump Analyzer subcommands available to the user, or detailed parameter information about a specified Dump Analyzer subcommand.

NOTE

This subcommand, as described below, is only available on NOS. If you are using the Dump Analyzer under NOS/VE, HELP invokes the SCL command that accesses online manuals.

Format

HELP
COMMAND_NAME = name
OUTPUT = file_name
STATUS = status_variable_name

Parameters

COMMAND_NAME (CN)

This parameter specifies the name of the Dump Analyzer subcommand for which parameter information is to be displayed. The specified value may be the full name of a Dump Analyzer subcommand, or any of its abbreviations. If the specified subcommand name cannot be found, a warning message is issued with no further display.

If the COMMAND_NAME parameter is omitted, then a display listing all available Dump Analyzer subcommands is produced (this is equivalent to entering DISCL,DO=B).

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is not specified, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand displays a list of all currently available Dump Analyzer subcommands:

```
HELP
```

Output from this subcommand is formatted as follows:

```
ANACD COMMAND LIST
------------------

DISPLAY_AUTO_DUMP_TABLE              DISPLAY_BOARD_MAP_TABLE
DISPLAY_BUFFER_CHAIN                 DISPLAY_CALLS
DISPLAY_COMMAND_INFORMATION          DISPLAY_COMMAND_LIST
DISPLAY_DATA_QUEUE                   DISPLAY_DI_SYSTEM_STATUS
DISPLAY_EXECUTIVE_ERROR_TABLE        DISPLAY_HARDWARE_STATUS
DISPLAY_LINKED_LIST                  DISPLAY_LINE_CONTROL_BLOCK
DISPLAY_LOG_QUEUES                   DISPLAY_MEMORY
DISPLAY_MEMORY_HEADER                DISPLAY_MEMORY_MAP
DISPLAY_MEMORY_USERS                 DISPLAY_NETWORK_STATUS
DISPLAY_PROGRAM_VALUE                DISPLAY_SYSTEM_CONFIG_TABLE
DISPLAY_TASK_CONTROL_BLOCK           DISPLAY_TREE
HELP                                 QUIT
SELECT_TASK                          VALIDATE_GLOBAL_INFORMATION
VALIDATE_STACK_AREAS
```

The following subcommand displays information about the DISPLAY_MEMORY subcommand:

```
HEL DISM
```

Output from this subcommand is formatted as follows:

```
ADDRESS, A        : any = $required
BYTE_OFFSET, BO   : integer  0 .. 7FFFFFFF(16) = 0
BYTE_COUNT, BC    : integer  1 .. 7FFFFFFF(16) = 1
REPEAT_COUNT, RC  : integer  1 .. 7FFFFFFF(16) or key ALL = 1
OUTPUT, O         : name = $optional
STATUS            : name = $optional
```

QUIT (QUI)
ANACD Subcommand

Purpose This subcommand terminates the Dump Analyzer session and returns control to the caller or operating system (NOS/VE or NOS).

Format

QUIT
STATUS = status_variable_name

Parameters *STATUS*

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks Under NOS/VE, an end-of-information (EOI) on an input file produces the same result as the QUIT subcommand does interactively. Under NOS, an EOI, end-of-record (EOR), or end-of-file (EOF) on an ANACD input file produces the same result as the QUIT subcommand does interactively.

SELECT_TASK (SELT)
ANACD Subcommand

Purpose This subcommand specifies the task that is to be examined in subsequent DISPLAY_CALL or DISPLAY_TASK_CONTROL_BLOCK subcommands. If SELECT_TASK is not used to specify a task, the task that was running when the DI was reset is assumed as the current task.

Format

```
SELECT_TASK
    TASK_IDENTIFIER = address or task_name
-- STATUS = status_variable_name
```

Parameters *TASK_IDENTIFIER (TI)*

If this parameter specifies an address, then it is interpreted as the first byte address of the Task Control Block (TCB) for the task selected to be the current task. If a task name is specified, then a TCB is sought with that name; if more than one TCB is found for that task, a warning is issued and there is no change in the current task.

NOTE

The only unique identification of a task for dump analysis purposes is the TCB address. A task name (taken from the module name when the task is scheduled) might be associated with more than one TCB. An attempt to select a task by task name is rejected if the task name is not unique.

If no TASK_IDENTIFIER is specified, the task that was running at the time the DI was reset is made the current task. If there was no task running when the DI dumped its memory, then Dump Analyzer error message number 94 is issued.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand causes the task with its TCB at address 10FFFE(16) to be selected as the current task:

```
SELECT_TASK TASK_IDENTIFIER=10FFFE
```

The following subcommand causes the task EXEC_MPB (provided it has only one scheduled instance in the dump) to be selected as the current task.

```
SELT TI=EXEC_MPB
```

The following subcommand causes the task that was executing when the DI was reset to be selected as the current task (provided there was a task running when the DI dumped its memory):

```
SELT
```

VALIDATE_GLOBAL_INFORMATION (VALGI) ANACD Subcommand

Purpose This subcommand displays general information that can serve as a preliminary guide to further analysis. This information includes suspended tasks, tasks with overwritten stacks, modules with code present in a dump, and tasks in excess of 30 ITMs queued. It also validates memory chains and task chains.

Format

VALIDATE_GLOBAL_INFORMATION
OUTPUT = file_name
STATUS = status_variable_name

Parameters *OUTPUT (O)*

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Remarks Execution time for this subcommand is typically longer than for other subcommands. Display is output slowly.

Examples The generic format of VALGI output is as follows:

```
VALIDATE GLOBAL INFORMATION

Reset code = xx : xxxxxxxxxxxxxxxxx
Date and time of reset       = yy/mm/dd hh.mm.ss.msec
Date and time of last reload = yy/mm/dd hh.mm.ss.msec
Running task = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, TCB address = xxxxxxxx(6)
<or 'No task was running at time of reset'>

Abort message:
<String from abort_system if not blanks>

Suspended Tasks:
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)   <task name / TCB addrss>
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)     OR  'none'

Task(s) with overwritten stack:
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)   <task name / TCB addrss>
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)     OR  'none'

Module(s) with code present in dump - possible invalid checksum:
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx                 <module name> OR  'noe'
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Task(s) with excess of 30 ITM's queued:
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)   <task name / TCB addrss>
  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx / xxxxxxx(16)     OR  'none'
```

The following subcommand writes VALGI output to file VALGOUT:

```
VALIDATE_GLOBAL_INFORMATION OUTPUT=VALGOUT
```

Display from this subcommand is formatted as follows:

```
VALIDATE GLOBAL INFORMATION

Reset code = 32 : KILL_SYSTEM_WITH_DUMP
Date and time of reset       = 86/11/21 15.02.53.700
Date and time of last reload = 86/11/21 15.01.51.333
Running task = EMAL_TEST                    , TCB address =   139244(16)

Abort message:
emal7 release descriptor failed

Suspended Tasks:
  None

Task(s) with overwritten stack:
  None

Module(s) with code present in dump - possible invalid checksum:
  None

Task(s) with excess of 30 ITMs queued:
  None
```

NOTE

Ordinarily, a DI code module (any module built for READ or READ/EXECUTE only) is written to the dump file only if it is found to have a bad checksum. However, when checksum information for modules is not available, all memory is dumped.

In this situation, the list of code modules in the dump file can be quite long, even though the listed modules do not necessarily have bad checksums.

VALIDATE_STACK_AREAS (VALSA) ANACD Subcommand

Purpose

This subcommand checks the integrity of the user stack and reserved stack areas and indicates whether or not these areas have been written into.

VALSA also checks the Task Control Block (TCB) chain and reports the first address at which the chain is broken. If a break is detected, VALSA then searches all of system main memory (SMM) and reports the addresses at which the TCB identifier (!TCB) was found.

Format

VALIDATE_STACK_AREAS
TASK_IDENTIFIER = address, task_name, or *keyword*
DISPLAY_OPTION = keyword
OUTPUT = file_name
STATUS = status_variable_name

Parameters

TASK_IDENTIFIER (TI)

If this parameter specifies an address, then it is assumed to be the address of a TCB. VALSA validates the stack areas associated with that TCB. If this parameter specifies a task name, all TCBs scheduled for that task are processed. The keyword ALL causes VALSA to process the entire TCB chain, beginning with the system ancestor task.

If TASK_IDENTIFIER is omitted, VALSA processes the entire TCB chain, beginning with the system ancestor task.

DISPLAY_OPTION (DO)

This parameter specifies the extent of information to be displayed. If the keyword FULL (or EXPANDED) is specified, all stack penetration data for the TCBs specified by the TASK_IDENTIFIER parameter are displayed. If BRIEF (or SUMMARY) is specified, then only a summary of the stack areas of questionable integrity is displayed.

If DISPLAY_OPTION is omitted, then the value BRIEF is assumed.

OUTPUT (O)

This parameter specifies the name of the file to which the subcommand output is written. This file may be any valid path name on NOS/VE. Under NOS, this file is local.

If OUTPUT is omitted, the output file name on the ANACD command is assumed.

STATUS

This parameter is the standard NOS/VE SCL status variable. Under NOS, the STATUS parameter is checked for correct syntax but is not operational.

Examples The following subcommand makes all stack area checks and writes output to file MYLIST.

```
VALIDATE_STACK_AREAS DISPLAY_OPTION=FULL OUTPUT=MYLIST
```

Display from this subcommand is formatted as follows, with one stack usage and penetration summary displayed for each TCB in the TCB chain:

```
STACK USAGE AND PENETRATION SUMMARY

  TASK CONTROL BLOCK ADDRESS                  13BE8A(16)
  TASK NAME                                SYSTEM_ANCESTOR
  STACK BEGINS AT ADDRESS (LOW  MEMORY)       13B57C(16)
  STACK ENDS   AT ADDRESS (HIGH MEMORY)       13BE6C(16)
  RESERVED STACK AREA
     SPACE ALLOCATED                             F0(16) BYTES
     STARTING ADDRESS ( LOW MEMORY)           13B57C(16)
     ENDING ADDRESS   (HIGH MEMORY)           13B66B(16)
  USER STACK AREA
     SPACE ALLOCATED                            8F0(16) BYTES
     STARTING ADDRESS ( LOW MEMORY)           13B66C(16)
     ENDING ADDRESS   (HIGH MEMORY)           13BE6B(16)
  STACK PENETRATION DATA
     PENETRATION DEPTH                          1D7(16) BYTES
     STARTING ADDRESS (HIGH MEMORY)           13BE6B(16)
     ENDING ADDRESS   ( LOW MEMORY)           13BC94(16)
```

The following message appears before the stack usage and penetration summary for a task that has overflowed its stack area:

```
OVERFLOW IN RESERVED STACK AREA AT ADDRESS  xxxxxxx
```

where xxxxxxx is the hexadecimal address at which the overflow occurred.

The following subcommand writes a summary of all stack areas of questionable integrity to file MYLIST:

```
VALSA DO=B O=MYLIST
```

Output from this subcommand appears in one of two forms. If no corruption of the reserved stack area is detected, then the following message is issued:

```
Specified reserve stack areas have passed penetration checks.
```

The following display identifies the tasks with reserved stack area corruption.

```
SUMMARY LIST OF TASKS WITH STACK CORRUPTIONS
--------------------------------------------

Task Name                          TCB       Type of Stack Corruption
                                   Address
-------------------------------------------------------------------------

system_ancestor_task             | 103457  | Overflow area penetrated
cmd_display_transport_status     | 1A0093  | Overflow area penetrated
itm_timer                        | 1C7300  | A6 out of range
update_nsl                       | 2AAA00  | Overflow area penetrated
```

NOTE

Error messages (if any) are dispersed throughout the display and identify, by TCB address, the task(s) with errors.

The following subcommand displays summary stack penetration information regarding any TCBs associated with the task named ITM_TIMER.

```
VALSA  TI=ITM_TIMER
```

If no corruption of the reserved stack(s) for ITM_TIMER task(s) is detected, the following message is displayed:

```
Specified reserve stack areas have passed penetration checks.
```

Display from this subcommand is formatted as follows if a corruption in the stack(s) of ITM_TIMER task(s) is detected:

```
SUMMARY LIST OF TASKS WITH STACK CORRUPTIONS
--------------------------------------------

Task Name                          TCB       Type of Stack Corruption
                                   Address
-------------------------------------------------------------------------

itm_timer                        | 1C7300  | A6 out of range
```

Appendixes

Glossary A

A

ADT

Refer to Auto Dump Table.

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.

American Standard Code for Information Interchange (ASCII)

A standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

ASCII

Refer to American Standard Code for Information Interchange.

Asynchronous 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 Dump Table (ADT)

A data table that is constructed when a device interface resets. The ADT is part of the dump file, and contains control information about the remainder of the file. The Dump Analyzer program uses the ADT to interpret the dump file.

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.

A protocol data unit (PDU) between the peripheral processor and the channel interface.

Board

Refer to Logic Board.

Boot File

On NOS, the boot file contains the basic set of software that is loaded into a device when the device interface requests to be loaded. A boot file brings a device interface up to a basic operational state. Further definition of the device interface's functions is provided by its system configuration file.

On NOS/VE, individual boot files do not exist. Rather, a set of boot software exists for each software version in the object library.

Buffer

One of two structures for the storage of data in device interface memory. See also Data Buffer and Descriptor Buffer.

Buffer Chain

If a message cannot be recorded into a single data buffer, additional data buffers are assigned and chained to the current buffer until the entire message is contained in the buffer chain.

Byte

1. (ISO) A binary character string operated upon as a unit and usually shorter than a computer word.

2. (ISO) 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

Refer to Concatenated Network.

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.

CCL

Refer to CYBER Control Language.

CDCNET

Refer to Control Data Distributed Communications Network.

CDNA

Refer to Control Data Network Architecture.

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.

Character

Any alphabetic, numeric, or special symbol that can be encoded. This term applies to the graphic characters for a terminal input or output device and to the encoded control characters used by the terminal. Within Control Data hardware, a character is a coded byte of data, such as a 6-bit display code (NOS only) or 7-bit ASCII code.

CIM

Refer to Communications Interface Module.

CLCB

Refer to Configured Line Control Block.

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.

Code Set Procedure

A CDCNET load procedure that allows a site to define its own code sets for input and output devices.

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.

Communications Interface Module (CIM)

The logic board within a CDCNET device interface that controls transmissions between the line interface module (LIM) bus and the internal system bus (ISB).

Computer Network

A linked collection of data processing and communications equipment.

Concatenated Network (Catenet)

A communications network composed of more than one type of communications medium (more than one network solution); often established when it is necessary to interconnect a local area network (LAN) with other resources (for example, another local area network, or geographically remote computer-related resources). Also called a catenet.

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 downline-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.

Configured Line Control Block (CLCB)

A data structure used by the device interface to maintain information about its defined communication lines.

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.

Connection-Oriented Network Service (CONS)

OSI connection oriented network service specification used in conjunction with X.25.

Connection Timeout

The service connection is discontinued after a specified length of inactivity.

CONS

See Connection-Oriented Network Service.

Control Data Distributed Communications'/ Network (CDCNET)

1. The collection of compatible hardware and software products offered by Control Data to interconnect computer resources into distributed communications networks.
2. A network that is interconnected by Control Data Network Architecture (CDNA)-compatible hardware and software products.

Control Data Network Architecture (CDNA)

The network architecture designed by Control Data. CDNA follows the lower layer recommendations of the International Standards Organization's (ISO) Open System Interconnection (OSI) reference model.

CPU

Refer to Central Processor Unit.

CRC

Refer to Cyclic Redundancy Check.

CYBER Control Language (CCL)

A language that allows you to insert a set of statements into the control statement section of your job. It also provides a means for you to determine the status of files, initiate tests and transfer within the control statement section, and display results in your dayfile.

CYBIL

Primary implementation language for NOS/VE computer systems and CDCNET software.

Cyclic Redundancy Check (CRC)

A check code transmitted with blocks of data. This code is used by several protocols.

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.

Data Circuit-Terminating Equipment (DCE)

1. In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line in a data station. The DCE may be separate equipment or an integral part of the DTE or of intermediate equipment. The DCE may perform other functions that are normally performed at the network end of the line.
2. The hardware that links data terminating equipment (DTE) to communications media. Data communications equipment is normally a modem or modem equivalent (data set).

Data Communication

The interchange of data messages from one point to another over communications channels.

Data Terminal Equipment (DTE)

1. That part of a data station which serves as a data source, data sink, or both.
2. Data communications equipment that allows human interaction with the databases and operations of a network.

DCE

Refer to Data Circuit-Terminating Equipment.

DCNS

Refer to Distributed Communications Network Software.

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 pre-selected value supplied for a missing parameter upon the entry of a command or subcommand.

Default Parameter Settings

The command parameter values to be assumed when no alternate values are supplied by the user.

Dependent Log ME

Refer to Log Management Entity.

Descriptor Buffer

A data structure used for chaining data buffers. Contrast with Data Buffer.

Device Interface (DI)

The communications processor that Control Data offers as its CDCNET hardware product. Also called a CDCNET device interface.

Device Interface Dump Analyzer

Host-resident software that processes device interface dump files.

Device Interface Resident Debugger (DIRD)

A software utility that enables software developers to debug their own CDCNET software. The DIRD is controlled from an interactive terminal connected to the device interface's main processor board (MPB).

DI

Refer to Device Interface.

DI Name Resolver

A program that resides in a DI and provides an interface between the DI and domain name servers. If a TCP/IP user specifies a domain name, the name resolver requests a domain name server to translate the name into the corresponding IP address.

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.

Diagnostic Management Services (DMS)

The device interface maintenance software that provides support to the device interface software and the online diagnostics.

Dial-up Line

A communications circuit created by dialing a destination over a common carrier's switched lines.

DIRD

Refer to Device Interface Resident Debugger.

Disabled

Cannot be used for normal network operation. Applies to boards, communication lines and network solutions.

Distributed Communications Network Software (DCNS)

The software that executes in a device interface as part of the CDCNET product.

DMS

Refer to Diagnostic Management Services.

DMTO

Refer to Deadman Timeout.

DOD

Department of Defense.

Domain Label

Part of a domain name and contains up to 63 characters. The label must begin with a letter (A..Z or a..z), which can be followed by letters, digits, or hyphens. The label must end with a letter or a digit.

Domain Name

TCP/IP users typically use domain names instead of IP addresses to reference TCP/IP services. Domain names identify hosts or other resources connected to a TCP/IP network. A domain name consists of a sequence of domain labels, arranged in a hierarchical order, and separated by periods. For example, the name PINK.ARH.CDC.COM, could specify a machine called PINK at the Arden Hills (ARH) facility of Control Data, which is a commercial organization (COM). The length of a domain name including the separating periods can be up to 255 characters.

Domain Name Server

A program that resides in a host connected to the TCP/IP network and responds to queries for information about domain names.

Down

A status of suspended service.

DTE

Refer to Data Terminal Equipment.

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. Refer to Analyze_CDCNET_Dump (ANACD).

E

Entry Point

A machine address corresponding to the start of the executable code in a software routine. Each externally declared (XDCLed) variable in the CYBIL source code of the software running in a device interface has a corresponding entry point in device interface memory.

ESCI

Refer to Ethernet Serial Channel Interface.

Ethernet

A baseband local area network protocol developed by the Xerox Corporation. CDCNET supports an Ethernet-compatible network.

Ethernet Serial Channel Interface (ESCI)

The logic board within a CDCNET device interface that controls transmissions between an Ethernet (IEEE 802.3) transceiver and the internal system bus (ISB) of the device interface.

Exception List

A file that determines how to process the load requests of the network's device interfaces (DIs). The exception list is a file of commands that specify the version of software to be loaded into the device interface, and which error codes should trigger a dump of the device interface memory. There is one exception list for the network, containing a default entry and any exceptions to the default entry.

F

Forms Code

A 1- through 6-character identifier associating a print file with a certain printer form ensures output will be routed to a printer which 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.

G

Gateway

A software interface between systems with different architectures and protocols.

H

HASP

Refer to Houston Automatic Spooling Program.

HDLC

Refer to High-Level Data Link Control.

High-Level Data Link Control (HDLC)

The International Standards Organization's (ISO) bit-oriented protocol for the data link layer of the Open Systems Interconnection (OSI) reference model.

Host Computer

A mainframe computer system, connected to a communications network, which 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.

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.

ICA Channel Interface (ICI)

The ICI is the I/O subsystem in the ICA that interfaces to a CYBER 180 channel.

ICA Ethernet Interface (IEI)

The IEI is the I/O subsystem in the ICA that interfaces to the Ethernet.

ICI

Refer to ICA Channel Interface.

IEI

Refer to ICA Ethernet Interface.

IFA

Refer to Independent File Access.

Independent File Access (IFA)

1. A service in some device interfaces that may access files in secondary storage on NOS host computers with the help of the host application called File Server. Only mainframe device interfaces (MDI) or mainframe terminal device interfaces (MTI) may contain an Independent File Access service.

2. A service on NOS/VE host computers that provides access to files in secondary storage.

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.

Information Processing Family Version 2 (IPF2)

A CDC host product used by CDCNET to generate Network Performance Analyzer (NPA) reports.

Initialization Procedure

A CDCNET load procedure that defines data to be sent to a printer when the printer's communication line becomes active.

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 series host computer mainframe.

Interface

A mechanism that enables the exchange of data between two dissimilar resources in a communications network.

International Standards Organization (ISO)

A worldwide standards group similar in function to the American National Standards Institute (ANSI). ANSI is a member of International Standards Organization.

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.

IP Address

Internet Protocol (IP) uses a 32-bit IP address field containing the Internet Address. Each IP system or host in the DDN is assigned a unique IP address. A host may have one or more IP addresses; however, a CDC CYBER host basically supports only one IP address per host.

IPF2

Refer to Information Processing Family Version 2.

ISO

Refer to International Standards Organization.

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

LAN

Refer to Local Area Network.

LCM

Refer to Line Control Module.

LCSM

Refer to Line Control Support Module.

LIM

Refer to Line Interface Module.

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.

Line Control Module (LCM)

CDCNET software that processes line and terminal configuration commands, and monitors line and terminal activities.

Line Control Support Module (LCSM)

CDCNET software that enables communications lines and performs autorecognition.

Line Interface Module (LIM)

A smaller logic board within a CDCNET device interface that enables the device interface to be attached to terminal, workstation, and unit record equipment lines.

Link

1. Any specified relationship between two device interfaces in a network, or a communication path between two device interfaces, or a data link.
2. The communications path between two device interfaces. Also called a line, channel, or circuit.

Load Procedure (LP)

A file containing commands specifying information to be downloaded to a printer. Load procedure types include initialization procedures (IPs), file prefix procedures (FPPs), code set procedures, and VFU load procedures (VLPs).

Local Area Network (LAN)

A privately owned communications network that interconnects computer-related resources. Typically, the resources interconnected by this network are confined to a relatively concise geographic area, such as a single building.

Log File

A file that is created and maintained by the operating system for storing error information and usage data concerning network elements.

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 ME

Refer to Log Management Entity.

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 Link Control 2 (LLC2)

OSI connection-oriented data link protocol utilized by CONS/X.25 when running over ESCI.

Logical Name

A name assigned to a CDCNET component (device interface, network solution, communication line, gateway) in the logical definition of the network. Many network operations commands refer to CDCNET components by their logical names. Contrast with Title.

Logical Unit (LU)

A 3270 terminal device from which a 3270 terminal interface program (TIP) accepts an interactive session.

LP

Refer to Load Procedure.

LU

Refer to Logical Unit.

M

Main Processor Board (MPB)

The logic board within a CDCNET device interface that provides the primary processing power for the device interface.

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 CDCNET device interface variant that interconnects a 12-bit channel of host computers operating under NOS or NOS/VE with an Ethernet (IEEE 802.3) local area network.

Mainframe/Terminal Device Interface (MTI)

The CDCNET device interface variant that interconnects 12-bit NOS and NOS/VE host computers with terminals, workstations, and unit record equipment without requiring a local area network.

Maintenance Host

In CDCNET, a NOS or NOS/VE host responsible for collection of all network errors and production of error analysis reports.

Maintenance Software

Software designed to perform system tests and diagnostics. All CDCNET maintenance software is onboard and online.

Manage CDCNET Configuration (MANCC) Utility

A CDCNET host utility for NOS that helps create, edit, and display CDCNET configuration files.

Management Data Unit (MDU)

A generic data structure used within protocol data units for the expression of variable data types.

Management Entity (ME)

CDCNET software that performs network management functions. CDCNET supports various MEs to perform specific network tasks.

MANCC

Refer to Manage CDCNET Configuration Utility.

MCI

Refer to Mainframe Channel Interface.

MDI

Refer to Mainframe Device Interface.

MDU

Refer to Management Data Unit.

ME

Refer to Management Entity.

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.

Memory Map

A listing of loaded software modules and their locations in device interface memory.

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.

MPB

Refer to Main Processor Board.

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.

NETCU

Refer to Network Configuration Utility.

NETDIR

The NOS permanent file name of the CDCNET file directory. NETDIR resides under user name NETOPS.

NETFM Utility

Refer to Network File Manager Utility.

NETLS

Refer to Network Log Server.

NETOPS

A NOS user name under which files are stored for use during CDCNET installation and by CDCNET-host operations. NETOPS contains files created and written by NAM while NAM is operating, the network directory file (NETDIR), and the NAMSTRT procedure.

NETOU

Refer to Network Operator Utility.

Network

An interconnected set of host computers, terminals, workstations, and unit record equipment. Refer also to Local Area Network and Concatenated Network.

Network Access Method (NAM)

The access method that resides under NOS; allows host-based network applications programs to exchange information with communications networks.

Network Access Method/Virtual Environment (NAM/VE)

The access method that resides under NOS/VE; allows host-based network applications programs to exchange information with communications networks.

Network Configuration Utility (NETCU)

A CDCNET utility on NOS/VE that logically configures CDCNET.

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 File Manager (NETFM) Utility

A CDCNET host utility that manages the CDCNET file directory. NETFM allows a user to do such activities as creating directory entries for CDCNET files, attaching CDCNET files, reviewing the directory, and deleting, updating, and verifying directory entries.

Network File System (NFS)

A software product of Sun Microsystems, Inc. that allows a variety of machines and operating systems to share files.

Network Identifier

A unique identifier (32-bit character string) assigned to a network solution.

Network Invocation Number (NIN)

A one- through three-digit decimal number that is incremented by one every time NAM is brought up. The NIN is used in NOS permanent file names for device interface dump files.

Network Job Entry Facility (NJEF)

The network applications software that supports IBM's Network Job Entry (NJE) protocol on NOS.

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 Operating System (NOS)

The software that controls data processing and storage in a CYBER 170 mainframe or a CYBER 180 mainframe (running NOS only or dual-state). CDCNET files stored and processed in CYBER 170 mainframes, such as configuration and boot files, network log files, and CDCNET host applications, are run under the Network Operating System.

Network Operating System/Virtual Environment (NOS/VE)

The software that controls data processing and storage in CYBER 180 mainframes. CDCNET files stored and processed in CYBER 180 mainframes, such as configuration and boot files, network log files, and CDCNET host applications, are run under the Network Operating System/Virtual Environment.

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 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 Service Access Point Address (NSAP Address)

An address used in the OSI protocol stack that uniquely identifies a CDCNET system and a user of the OSI Network layer within that system. An NSAP address consists of two parts: a Network Entity Title and an NSAP selector. The Network Entity Title uniquely identifies a CDCNET system. The NSAP selector uniquely identifies a user of the OSI Network layer in that system.

Network Solution

A communications medium over which data is transmitted between interconnected network resources, and which uses CDCNET protocols. In OSI terminology, a network solution is also referred to as a subnet. 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.

Network Transfer Facility (NTF)

An application providing a fully symmetric queue file transport facility between a NOS/VE host and another host in a geographically dispersed network. NTF supports IBM's Network Job Entry (NJE) protocol and HASP multileaving protocol for communication between hosts.

NFS

Refer to Network File System.

NIN

Refer to Network Invocation Number.

NJEF

Refer to Network Job Entry Facility.

NLTERM

Refer to Network Logfile Termination Utility (NLTERM).

NOS

Refer to Network Operating System.

NOS/VE

Refer to Network Operating System/Virtual Environment.

NPA

Refer to Network Performance Analyzer.

NSAP Address

Refer to Network Service Access Point Address.

NVT

Refer to TELNET Network Virtual Terminal.

O

Onboard

ROM-resident; for example, the self-test diagnostics in the device interface.

Online

Condition in which a network hardware or software component is present and active within the network configuration.

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).

Open System Interconnection (OSI)

The International Standards Organization's (ISO's) reference model for network processing. This model is based on a network architecture that segregates network functions into seven layers.

OSI

Refer to Open System Interconnection.

P

Passthrough

Refer to Terminal Passthrough.

PDN

Refer to Public Data Network.

PDU

Refer to Protocol Data Unit.

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.

PMM

Refer to Private Memory Module.

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.

Printer Support Utility (PSU)

The network applications software that supports standalone CDCNET printers on NOS.

Private Memory Module (PMM)

An optional device interface board with 128K bytes of static RAM dedicated to the main processor board (MPB) for code execution.

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 pre-defined 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.

Protocol Data Unit (PDU)

A data unit that is used to communicate information between peer entities.

Protocol Stack

A collection of protocols in successive layers. CDCNET is based on ISO's Open System Interconnection (OSI) reference model, where each system includes a set of layers and each layer supports one or more protocols.

CDCNET phase 2 of OSI support includes support for OSI and TCP/IP protocols for layers 3 and 4. Therefore, CDCNET supports two protocol stacks: OSI and TCP/IP.

PRU

Refer to Physical Record Unit.

PSR

Refer to Programming System Report.

PSU

Refer to Printer Support Utility.

Public Data Network (PDN)

A commercial packet-switching network that supports the communications interface described in CCITT protocol X.25.

R

RBF

Refer to Remote Batch Facility.

Real-time Clock

A time-of-day clock maintained by CDCNET device interfaces.

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.

Relay

Process occurring when CDCNET receives a data unit from a directly connected network solution and transmits the data unit to another directly connected network solution.

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

SAP

Refer to Service Access Point.

SAP Identifier

A 16-bit number that uniquely identifies a SAP for other software components.

SCL

Refer to System Command Language.

SCT

Refer to System Configuration Table.

SDLC

Refer to Synchronous Data Link Control.

Segment

The unit of data exchanged by TCP modules. This term also describes the unit of exchange between any transport protocol modules.

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.

Service Access Point (SAP)

An exchange point between the services of two adjacent Control Data Network Architecture (CDNA) layers.

Single-Bit Error

A transmission error that affects only one bit in a block of data. Single-bit errors can be detected, isolated, and corrected automatically.

Site Administrator

A person who configures a site's network, supervises hardware and software changes, and ensures that the network operates efficiently.

SMM

Refer to System Main Memory.

SMM4

A 4 M byte version of the SMM (see System Main Memory).

SNA

Refer to Systems Network Architecture.

SNA3270 TIP

A terminal interface program that provides IBM 3270 Information Display System users access to CDCNET through an SNA network.

Socket

A TCP/IP address used to locate a process on a host. This address is used by Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). It consists of the 32-bit IP address and a 16-bit port number.

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).

Stack

An area in memory used as temporary storage for chaining calls during task or interrupt service routine execution. Task calls are chained on a user stack. Interrupt service routine calls are chained on a supervisor stack.

Stack Frame

The area within a stack that accommodates a single call.

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.

Subnet

The concept of a subnet is used both in OSI and TCP/IP. Therefore, there are two definitions of subnet.

1. OSI: In CDCNET, individual systems are connected to each other via different media such as Ethernet and HDLC. A medium connecting two or more CDCNET systems is called a network solution or a subnet. For OSI, a subnet refers to one and only one network solution.

2. TCP/IP: TCP/IP subnetting is a required Internet standard. Subnetting allows a configuration consisting of many physical networks to be addressed with a single IP network number. Each physical network is assigned a subnet number. Each host is addressed based on its network number, subnet number, and host number. These three fields make up the IP address.

Subnet Identifier

An identifier that identifies a subnet in a CDCNET network. It must be unique, and must not be associated with more than one subnet in a CDCNET network.

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.

Synchronous Data Link Control (SDLC)

Bit-oriented data link control protocol developed by International Business Machines (IBM).

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, and configuration and terminal user commands are based.

System Configuration Table (SCT)

A data table maintained in device interface memory that contains the values of current and configurable system variables.

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 Operator Utility (NETOU). 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 containing 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.

Systems Network Architecture (SNA)

IBM standard defining the layers and layer protocols to be used within an IBM network.

SYSTEMX

A NOS user name that is used to store files for NOS and CDCNET installation and operations.

T

T-to-A

Refer to Terminal-to-Application.

TAM

Refer to Total Allocatable Memory.

Task Control Block (TCB)

The task control block is a data structure that maintains the context of a task while it is executing.

TCB

Refer to Task Control Block.

TCP

Refer to Transmission Control Protocol.

TCP/IP

Refer to Transmission Control Protocol/Internet Protocol.

TDI

Refer to Terminal Device Interface.

TDP

Refer to Terminal Definition Procedure.

TELNET Network Virtual Terminal (NVT)

A TCP/IP protocol that provides presentation layer services for other application protocols. TELNET NVT protocol is roughly equivalent to VTP in the ISO model. It establishes connections and controls interactive virtual circuits.

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 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 async, HASP, and IBM 3270) to communicate in CDCNET networks.

Terminal Passthrough

A CDCNET feature that allows interactive asynchronous terminal traffic to pass through the network transparently. The hosts and terminals interface with each other as if they were directly connected. Terminal passthrough allows a CDCNET-connected terminal user to access non-CDCNET supported hosts, such as NOS/BE and VAX.

Terminal-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.

Text

A sequence of characters forming part of a transmission that is sent from a source to a receiver, containing the information to be conveyed.

TIP

Refer to Terminal Interface Program.

TIP Interface Record Table (TIRT)

A TDI data structure that maintains information about active TIPs.

TIRT

Refer to TIP Interface Record Table.

Title

A string of 1 through 255 ASCII characters that identify 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.

Total Allocatable Memory (TAM)

The amount of memory at the end of configuration that is available for dynamic memory or buffer allocation. This excludes the memory used by active tasks.

Transmission Control Protocol/Internet Protocol (TCP/IP)

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.

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.

Transport Layer

Open Systems Interconnection (OSI) layer 4. Provides end-to-end control of a communication session once the path has been established. It allows processes to exchange data reliably and sequentially, regardless of which systems are communicating.

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.

Unit Record Interface (URI)

A Line Interface Module (LIM)-type peripheral circuit board that interfaces with the LIM bus and is used with the Communications Interface Module (CIM). The URI provides an 8-bit parallel interface for the operation of character or line printer. The URI includes all necessary drivers, receivers, timing, and control circuitry to drive one printer at a time.

Upper Layer Protocols (ULP)

A collective term for layers 5, 6, and 7 of the Open System Interconnection (OSI) network reference model.

URI

Refer to Unit Record Interface.

User TELNET

Allows a CDCNET terminal to connect to a foreign host's interactive service using TCP/IP TELNET communications.

V

VE Interface

A channel between a NOS/VE host and an MDI or ICA-II that uses the OSI protocol stack.

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

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

The Consultative Committee of International Telephone and Telegraph (CCITT) standard for the interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) in an X.25 packet-switching network.

X.25 Asynchronous TIP

Also known as X.29 PAD, this is a CDCNET feature that allows asynchronous terminals to access CDCNET either by a Public Data Network (PDN) that supports the X.3 Packet Assembly/Dissassembly (PAD) facility or by the terminals operating in X.25 mode.

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 circuit. A-to-A connections over X.25 circuits are provided by the Network Products applications.

NLTERM, NLLIST, and NLPURGE Utilities (NOS Only) B

This appendix contains instructions, command descriptions, and messages generated by the following NOS utilities: Network Logfile Termination Utility (NLTERM), Network Logfile List Utility (NLLIST), and Network Logfile Purge Utility (NLPURGE). These utilities do not run under the Network Operator Utility (NETOU).

NLTERM (Network Logfile Termination Utility) Terminate CDCNET Log File

Purpose Invokes NLTERM. Terminates CDCNET network log files, and renames the terminated log files.

Format

NLTERM
OP = keyword value
NM = file name
L = file name

Parameters *OP*

Specifies the environment in which NLTERM and its parameters 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 are entered from host console.
T	Parameters are 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.

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 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 goes 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 receives 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 do not terminate the file, and 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 (Network Log File List Utility)
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. See the instructions for using NLTERM.

NLTERM has several subcommands.

Command	Description
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 (NM) 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.

Examples This example sends the list of terminated log files to a file called LOGFIL2.

```
nllist,l=logfil2.
```

NLPURGE (Network Logfile Purge Utility)
Purge Terminated CDCNET Log Files

Purpose Purges previously terminated network log files. The PURGE subcommand in the NLTERM interactive or K display interface performs the same function as NLPURGE.

Format

NLPURGE
NM = file name

Parameters

NM

Name of the terminated log file to purge (see format specified for the NM parameter used with NLTERM).

Remarks The PURGE command in the NLTERM interface performs the same function as NLPURGE>.

Examples This example purges the terminated log file NLA0101.

```
nlpurge,nm=nla0101
```

CDCNET File and Procedure Management Utilities (NOS Only) C

This appendix describes the following CDCNET NOS utilities: Network File Manager (NETFM) and related utilities (NETBDF, NETMDF, and NETRDF), and Network Procedure Library Manager (NETPLM). Messages generated by these utilities are documented in the NOS Operations handbook and the NOS Reference Set, Volume 3.

Network File Manager Utility (NETFM) Overview

Network File Manager (NETFM) is a NOS utility that manages CDCNET host files and maintains the CDCNET directory. This directory correlates CDCNET file names with their NOS permanent file names. When CDCNET configuration files are created and edited, you use NETFM to create or update their directory entries.

The CDCNET directory is a private, direct-access permanent file, named NETDIR. NETDIR has entries for all CDCNET files on the NOS host. It is the only link between NOS file names and CDCNET file names. When DIs are initialized, NETDIR determines NOS file names for CDCNET configuration files (exception list, boot files, all system configuration procedures, terminal definition procedures (TDPs), terminal user procedures (TUPs), and load procedures (LPs)). NETFM attaches and purges these CDCNET files, and creates, deletes, lists, updates, and verifies directory entries.

To create directory entries for files, NETFM executes the NOS commands CATLIST and ATTACH (in read mode) on files. In addition, NETFM can be used only from user names permitted to access NETDIR. Typically, NETFM executes from the user name in which the file is stored, so no special effort is needed to access NETDIR, or to allow NETFM to execute ATTACH and CATLIST. However, if a file is under an alternative user name, you must allow ATTACH and CATLIST on the file and make it available to the user name on which NETFM executes. To allow CATLIST on the file by NETOPS, NETADMN, or another user name, specify AC=Y on the DEFINE, SAVE, REPLACE, or CHANGE command for the file. To allow user names to access NETDIR, and to specify read access for files, use the PERMIT command.

```
PERMIT,filename,username=mode (READ)
```

NOTE

When adding directory entries with NETFM, the NETFM job must execute under a user name. Otherwise, added entries are invalid (because there is no user name in the user name field in the directory entry), causing file access attempts to fail. A no-user-name situation can occur when running NETFM at the system console, and when X.DIS and SUI are specified to access an account. To remedy this situation, use a user name statement instead of SUI to access the account where NETFM runs. This recommendation also applies to NETMDF.

Frequently accessed CDCNET files, such as TDPs and TUPs, should have the retain flag (RT=Y) set in their directory entries (see RT description in the CREATE directive in this appendix). This causes Network File Server (NETFS) to keep an accessed file local during a retain timer period. If the file is accessed again during that period, NETFS does not have to reattach the file, saving time and file use.

NETFM Directives Summary

Use the directives described in table C-1 to maintain the CDCNET directory and to access CDCNET files. More detailed descriptions of the directives are provided later in this appendix.

Table C-1. NETFM Directives Summary

Directive	Description
ATTACH	Gets/attaches a permanent file based on information in the directory.
CREATE	Creates an entry in the CDCNET directory.
DELETE	Deletes an entry in the CDCNET directory. Does not purge the permanent file associated with the directory entry.
LIST	Lists a CDCNET directory entry. You may list a single entry or all entries in the directory.
PURGE	Purges a NOS permanent file and deletes the file's directory entry.
UPDATE	Updates the CDCNET directory to match the NOS permanent file base.
VERIFY	Verifies that the CDCNET directory entry matches the characteristics of the actual file.

Three procedures help you use NETFM directives: ATTCF (Attach Configuration File), DEFCF (Define Configuration File), and REPCF (Replace Configuration File). These procedures execute FCOPY and NETFM directives for NOS CDCNET configuration files.

When executing CREATE, UPDATE, and VERIFY, NETFM internally gets/attaches permanent files specified in the PF (permanent file) parameter using local file ZZZZZFM. To prevent file-busy errors, return all direct access files in write mode to be accessed by CREATE, UPDATE, and VERIFY before accessing NETFM.

Using NETFM

1. Make sure the directory is already built. It should have been built during installation. If it is not, or if a new directory must be built, use NETBDF, described later in this appendix.

2. To run NETFM interactively, log in to IAF on a user name validated to access NETDIR.

3. Enter the following:

   NETFM,I=input file,L=list file,UN

   All parameters are optional. However, to use the official NETDIR on NETOPS, you must specify UN as a keyword. See the parameter descriptions in the NETFM command description later in this appendix.

4. Enter NETFM directives. See instructions for entering NETFM directives later in this appendix.

NETFM Command

Purpose Accesses Network File Manager (NETFM), which allows you to access the CDCNET directory, using NETFM directives. You use NETFM any time you create, attach, or purge files used by CDCNET. You also use NETFM to attach, create, delete, list, purge, update, and verify CDCNET directory entries.

Format

NETFM
abort option
I=input file
L=list file
UN=user name
z option

Parameters *abort option*

Abort option, specified by the keyword ABORT. If specified, the abort option causes NETFM to abort on nonfatal errors and not execute any further directives.

I=input file

The input file in which NETFM directives are located. You can specify this parameter using one of the following options:

Value	Description
I	Directives are on file INPUT.
I=lfn_1	Directives are on local file lfn1 (file name of your choice).
I=0	No input directive file. If this option is chosen, the z option must be specified in the NETFM command and in directives following the NETFM command terminator.
I (omitted)	Directives are on file INPUT or, if z option is specified, after the NETFM command.

L=list file

The output file in which NETFM list output is to be written. You can specify this parameter using one of the following options:

Value	Description
L=lfn2	List output in local file lfn2 (filename of your choice).
L=0	No list output.
L (omitted)	List output on local file OUTPUT.
L	List output on local file OUTPUT.

UN=user name

The user name from which the directory should be attached. You can specify this parameter using one of the following options:

Value	Description
UN	Attach the directory from user name NETOPS.
UN=user_name	Attach the directory from the user name specified.
UN (omitted)	Attach the directory from the user name where NETFM is executing.

z option

Specifies whether or not the NETFM command contains input directives (such as ATTACH, CREATE, and UPDATE). If Z is present, the input directive is taken from the command following the terminator of the NETFM call. If Z is omitted, the NETFM command does not contain input directives.

If Z is present and directives are present, the NETFM command has the following format:

NETFM,UN,Z./$input_1$/$input_2$ /...$input_n$

input	Specifies one line of NETFM directive input.
/	Separates input lines. This character must immediately follow the right parenthesis or period.

The total input line including the NETFM command, parameters, and directives, cannot exceed 80 characters, and continuation lines are not allowed. Continuation lines are allowed if NETFM directives are in a directive file (see Continuing NETFM Directive Entry on Another Line later in this appendix).

The following example specifies two LIST directives following the Z parameter on the NETFM command. The LIST directives list a directory entry for a CDCNET object library file with CDCNET file name MODULE#0102, and also lists all version 0102 boot files.

```
NETFM(Z)/LIST,LO=F,NF=MODULE#0102*/L,NF=BOOT#0102
```

Using NETFM Directives

This section contains instructions on using NETFM directives.

NETFM Directive Input Requirements

Interactive directives are always processed as 6-bit (normal), because NETFM has an embedded CSET (NORMAL) that overrides the terminal's current character mode. The initial character mode is restored when NETFM terminates.

CAUTION

Do not enter a user break sequence while a NETFM directive is executing. Interruption of directive processing may cause errors in the directory, which later cause NETFM to malfunction. You can partially recover the directory using NETMDF and NETRDF (see descriptions later in this appendix).

The following example deletes the directory entry for a CDCNET configuration file. Either the CDCNET file name or the NOS permanent file name may be specified with the directive. This example specifies the NOS permanent file name, CF0072. The directive response is also shown.

If PF is the only parameter specified, you must execute NETFM from the user name under which the file is stored. In this example, NETFM was executed from user name NETSUPP.

```
DELETE,PF=CF0072
```

Response:

```
          NETWORK FILE NAME PFN       TYPE DN UN      PN CREATION  LAST MOD
                              FS                      RT DATE/TIME DATE/TIME
                          COMMENT

DELETED CONFIGURATION#0   CF0072     IND   NETSUPP  0 85/07/10. 85/07/10
        80025100072         1                       F 15.36.49. 15.36.49
```

The fields in a NETFM directive response are described below.

Field	Description
response	Indicates action taken after the NETFM directive was entered. In the example, DELETED is the response.
NFN	CDCNET file name. This name spans multiple lines, if necessary.
PFN	NOS permanent file name.
TYPE	File type, either direct (DIR) or indirect (IND) access.
DN	Device type of the auxiliary pack or device number of the family device on which file is stored. The device number may be from 1 through 77. Allowed device types are described in the NOS 2 Reference Set, Volume 3, in the section on permanent file commands.
UN	User name of the owner of the NOS permanent file.
PN	Pack name. The auxiliary device where the file is stored.
CREATION DATE/TIME	Date and time file was created. This information is not actually in the directory entry; it is obtained from a CATLIST of the file and combined with directory information.
LAST MOD DATE/TIME	Date and time file was last modified. This information is not actually in the directory entry; it is obtained from a CATLIST of the file and combined with directory information.
FS	File size, in decimal PRUs.
RT	Value of retain flag. Used only for files accessed by Network File Server (NETFS), which interfaces the CDCNET file access system with the NOS file access system. There are two possible values for RT: F Retain flag not set (default value) T Retain flag set
COMMENT	Brief description of the file or other comment (see Entering Comments in a Directory Entry, later in this appendix).

Specifying Directives on the NETFM Command

You may enter NETFM directives with the NETFM command itself, all on the same line, by using the Z option (see NETFM command format). A delimiter seperates multiple commands. If you specify NETFM directives following the NETFM command using the Z option, and if you execute NETFM as an interactive command, directives do not have to be in display code.

Example:

This example specifies two directives at the end of the NETFM command: DELETE (D) and LIST (L). When this command executes, first NETFM is accessed, then the CDCNET directory entry for NOS permanent file name CF0072 is deleted, and then all configuration procedures are listed (L,NF=CONF*). The slash character (/) separates the directives. The * at the end of the list directive is a wildcard character. See Wildcard Characters, described next, for more information.

```
NETFM(UN,Z)/D,PF=CF0072/L,NF=CONF*
```

Wildcard Characters

Wildcard characters provide flexibility when searching for CDCNET file names. You can use wildcard characters in file names on NETFM directives, rather than entering all file names matching the file name modified by wildcard characters. For example, wildcard characters specified in a file name on a LIST directive results in a listing of several directory entries for several files. Or, you can verify CDCNET directory entries for several files using one VERIFY directive and a CDCNET file name with wildcard characters.

The wildcard characters, described in table C-2, can be used with all NETFM directives except ATTACH and CREATE.

Table C-2. Wildcard Characters

Wildcard Character	Description and Use
?	Replaces any single character in a CDCNET file name except the underscore (_) or pound-sign (#) characters. In a search for CDCNET file names matching the name modified with the wildcard characters, the question mark matches any character except _ or # that appears where a question mark is part of the wildcard name. For example, MODULE#09?? matches all CDCNET file names beginning with MODULE#09 and ending with any two characters.
*	Represents any string of characters in a CDCNET file name. If the asterisk is the last character in the file name, it means the rest of the file name being compared to the wildcard character matches. For example, DUMP#FULL_* matches all CDCNET file names beginning with DUMP#FULL_. The asterisk wildcard character is useful when searching the directory for dump files or other groups of files. If the asterisk is not the last character, it matches zero or more characters up to the underscore or pound-sign character. For example, DUMP#*_*A2_* matches all dump files for a DI with a system ID ending in A2. If the asterisk is in the middle of a file name component or level (at neither the beginning nor the end of a name level), that level name must match the characters immediately preceding and following the asterisk. The asterisk matches zero or more characters. No wildcard characters can immediately follow an asterisk.
[X..X]	Matches a single character from the set in brackets. For example, BOOT#060[0..5]* matches any CDCNET boot file having a version level beginning with 060 and ending with any value from 0 through 5.
[^X..X]	Matches a single character that is not from the set in brackets.

Examples:

The following LIST directive lists all entries where the first three letters of the CDCNET file name are CON. Full directory entries are to be listed; this is specified by the LO parameter.

```
LIST NF=CON*, LO=F
```

Output for the LIST directive is as follows:

```
CONFIGURATION#0  CF0081   IND     NETADMN 0       85/10/28. 85/10/29.
80025100081             8                 F       21.03.49. 07.01.48.

CONFIGURATION#0  CF00A2   IND     NETADMN 0       85/10/28. 85/10/29.
800253000A2             8                 F       21.03.53. 07.00.43.

CONFIGURATION#0  CF00C0   IND     NETADMN 0       85/10/28. 85/10/29.
800253000C0             3                 F       21.03.54. 06.57.48.

CONFIGURATION#0  CF00BE   IND     NETADMN 0       85/10/28. 85/10/29.
800253000BE             3                 F       21.03.53. 06.57.42.
```

The following example uses the ? wildcard character to search for all CDCNET file names beginning with MODULE#0.

```
LIST NF=MODULE#0???
```

Output:

```
MODULE#0606

MODULE#0607
```

The following example uses the * wildcard character to search for all files beginning with BO (boot files).

```
LIST NF=BO*
```

Output:

```
BOOT#0606_ESCI

BOOT#0607_MCI
```

The following example uses the * wildcard character to search for all dump files listed in the directory.

```
LIST NF=DUMP#FULL*
```

Output:

```
DUMP#FULL_080025100080_850101081925

DUMP#FULL_080025100080_850101082538

DUMP#FULL_080025100080_850101082410
```

Continuing NETFM Directive Entry on Another Line

To continue a NETFM directive on another line, use the ellipsis (..) at the end of a line you have entered.

Example:

This example creates a directory entry for configuration procedure CONFIGURATION#080025100072 in the catalog of user name NETSUPP.

```
CREATE,PF=CF0072,UN=NETSUPP,NF=CONFIGURATION#080025100072
```

Entering Comments in a Directory Entry

You may use comments in a CDCNET directory entry to explain the file's contents. Both CREATE and UPDATE allow you to enter comments in the directory entry. For both directives, specify comments using the C parameter. Enter comments as a string value enclosed by quotation marks ("). If the comment itself contains quotation marks, the comment entry must have two consecutive quotation mark characters to indicate the quotation mark within the comment.

Example:

The following CREATE directive creates a new directory entry for a boot file. The entry contains a comment about the file. The comment itself contains quotation marks, so two consecutive quotation marks must be used in the comment string.

```
CREATE,PF=BT24DTE,NF=BOOT#0101_ESCI,UN=USERAA,..
C="TEST BOOT FILE FOR ""WEEK 1 TESTING"" PROGRAM"
```

ATTACH (A)

Purpose Attaches or gets a permanent file. You must be validated to access the permanent file with the requested access mode.

Format

ATTACH
LF=local file
PF=permanent file
NF=network file
UN=user name
PN=pack name
PW=password
M=mode

Parameters

LF=local file

Local file name used to reference the file.

PF=permanent file

The permanent file to attach. You do not also have to enter the network file name in order for ATTACH to execute. Either PF or NF is required. If PF and NF are specified, PF is ignored.

NF=network file

The CDCNET file name for the file. Either PF or NF is required. If NF and PF are specified, PF is ignored. For faster processing, use NF instead of PF whenever possible. Wildcard characters are not allowed on CDCNET file names on ATTACH.

UN=user name

User name under which the file is stored. If UN is specified, NETFM attempts to attach the file using a file entry matching the NF or PF and UN parameters specified. If you omit UN, the user name listed in the directory entry is used when NF is specified, and the current job's user name is used if PF alone (no NF parameter) is specified.

PN=pack name

Auxiliary device name. If PN is specified, the permanent file on the specified auxiliary pack is attached. The pack name must match the pack name stored in the directory entry. If PN=0, the permanent file residing on the system default family is attached. Default for PN is to ignore the pack name when searching the directory.

PW=password

Password required for attaching the permanent file.

M=mode

Access mode in which the permanent file is attached. The following access modes are allowed:

Value	Description
R	Read mode
W	Write mode

Default is R (read mode). This parameter has meaning only if the permanent file attached is a direct access file. For more information on file access modes, see the section on permanent file commands in the NOS Reference Set, Volume 3.

Examples This example attaches CDCNET file CONFIGURATION#080025100071 and writes it to local file CONFILE.

```
ATTACH,NF=CONFIGURATION#080025100071,LF=CONFILE

PERMANENT FILE ATTACHED.
```

This example attaches the DI dump file with NOS permanent file name DSAS323 and writes it to local file XX1.

```
A,PF=DSAS323,UN=NETOPS,LF=XX1

PERMANENT FILE ATTACHED.
```

CREATE (C)

Purpose Creates an entry in the CDCNET directory file, NETDIR. The NOS permanent file corresponding to the CDCNET directory entry must already exist. When executing CREATE for a file not stored under the user name where you are executing NETFM, you must be able to do CATLIST and ATTACH on the permanent file to create a directory entry. See the section on permanent file commands in the NOS Reference Set, Volume 3, for restrictions on CATLIST and ATTACH.

Format

CREATE
PF=permanent file
NF=network file
UN=user name
PN=pack name
R=device type
C=comment
RT=retain timer

Parameters

PF=permanent file

The permanent file name to be stored in the directory entry. If PF is the only parameter specified, you must execute NETFM from the user name under which the file is stored.

The file should be able to have a CATLIST performed on it by NETOPS, NETADMN, or any other user name under which you perform network administration activities. Specify these conditions by the parameter AC=Y on the NOS DEFINE, SAVE, REPLACE, or CHANGE command. If the file is private, it must have read permission permitted to NETOPS, NETADMN, or any other user name on which you perform network administration activities. Specify read permission using the PERMIT command.

PERMIT,filename,username=mode (READ)

NF=network file

The CDCNET file name to be stored in the directory entry. Wildcard characters are not allowed on CDCNET file names on CREATE.

UN=user name

User name under which the file is stored. If UN is specified, NETFM uses this user name when creating the directory entry. If UN is not specified, your current user name is used in the directory entry.

PN=pack name

Auxiliary device name to be stored in the directory entry. The permanent file is assumed to be on this pack. If PN is not specified or is zero, the default is the system default family.

R=device type

Device type (DT) identifier to be stored in the directory entry. Values allowed for this parameter are specified in Volume 3 of the NOS Reference Set, in the section on permanent file commands. The default for this parameter is the system default device.

C=comment

Comment text to be stored in the directory entry. Comments are specified as a string surrounded by quotation marks ("). Comment strings can range from 1 through 50 display code characters. Additional characters are truncated. If the string contains a quotation mark, two consecutive quotation marks must be used to designate a quotation mark in the comment text.

RT=retain timer

Value for the retain flag in the directory entry, which determines the directory access and NOS permanent file attachment process. This parameter is not normally needed for CREATE directives; it is meaningful only for files accessed by Network File Server (NETFS). Allowed keyword values are the following:

Value	Description
Y	Set the retain flag.
N	Do not set the retain flag.

Default is N.

If RT=Y, directory access and attachment of files are performed by NETFS only the first time inactive files are opened. Files remain attached by NETFS until one of the following conditions occurs:

- The network shuts down.
- NETFS terminates abnormally.
- The NETFS retain timer for a file expires.

If RT=N, NETFS accesses the directory and attaches a permanent file each time it receives a request to open a file while a file is inactive. To reduce performance overhead and central memory requirements, specify RT=Y for files NETFS frequently accesses, such as library files. RT=Y should be set for all files of type MODULE#.

The following table shows file types supported by NETFS, with the fixed portion of the CDCNET file name in parentheses for each file type, and the default retain timer values for each file type. A retain timer determines how long, after the most recent file access, NETFS retains a file before returning the file to the operating system.

CDCNET File Type	Default Retain Timer
Dump	0 seconds
Boot	5 seconds
Module library	180 seconds
Exception list	5 seconds
Configuration procedure (individual file)	5 seconds
Configuration procedure (library)	30 seconds
Load procedure (individual file)	30 seconds
Load procedure (library)	30 seconds
Terminal definition procedure (individual file)	30 seconds
Terminal definition procedure (library)	30 seconds
Terminal user procedure (individual file)	30 seconds
Terminal user procedure (library)	30 seconds
All others	0 seconds

NOTE

These default retain timer values can be changed by reinstalling the CHA (CDCNET Host Applications) product on NOS. See the NOS Installation handbook for details on CHA installation.

Remarks When executing CREATE, NETFM internally gets or attaches the permanent files specified by the PF parameter (using local file ZZZZZFM). To prevent file-busy errors, return all direct access files in write mode to be accessed by CREATE before accessing NETFM.

Examples The following examples are performed from user name NETADMN.

This example creates a new directory entry for CDCNET boot file BOOT#0535_ESCI. The boot file's NOS permanent file name is BTD0535. The file is stored under NETADMN with read permission to NETOPS.

```
CREATE,PF=BTD0535,NF=BOOT#0535_ESCI,C="TEST BOOT FILE NO. 3"
```

This example creates an entry for a terminal user procedure (TUP) USER_PROCEDURE#SAM. The TUP's NOS permanent file name is UPROCSM. UPROCSM is stored as a private file under user name JOEUSER, with read permission to NETADMN and NETOPS. Alternative users can perform a CATLIST on the file. The network administrator, on NETADMN, adds the TUP to the directory, since JOEUSER does not have (or need) permission to access and update the directory.

```
C,PF=UPROCSAM,NF=USER_PROCEDURE#SAM,UN=JOEUSER
```

DELETE (D)

Purpose Deletes an entry in the CDCNET directory file, NETDIR.

Format

DELETE
PF=permanent file
NF=network file
UN=user name
PN=pack name

Parameters *PF=permanent file*

The permanent file to delete. If PF is specified, the entry for the specified permanent file is deleted. Either PF or NF (CDCNET file name, below) is required. If both PF and NF are specified, PF is ignored. If only PF is specified, you must execute NETFM from the user name under which the file is actually stored.

NF=network file

The CDCNET file name. Either NF or PF is required to indicate which entry to delete. If both NF and PF are specified, PF is ignored. For faster processing, use NF instead of PF whenever possible. Wildcard characters are allowed in the CDCNET file names on DELETE.

UN=user name

User name under which the file is stored. If UN is specified, NETFM uses the user name specified. If you omit UN, the user name in the directory entry for the file is used. If you only specify PF, the user name in the directory entry must match the user name under which you are running NETFM.

PN=pack name

Auxiliary device name. If PN is specified, the permanent file on the specified auxiliary pack is deleted. PN must match the pack name stored in the directory entry. If PN is 0 (zero), the permanent file residing on the system default family is deleted. Default is to ignore the auxiliary device name when searching the directory.

Examples This example deletes the directory entry for an ESCI boot file, BOOT#0534_ESCI.

```
DELETE,NF=BOOT#0534_ESCI

DELETED BOOT#0534_ESCI  BT57AFA     IND DAT0214    84/03/25. 84/03/26.
                             11                F  09.35.01. 09.35.01.
                        TEST BOOT FILE NO. 2
```

LIST (L)

Purpose Lists a CDCNET directory entry or entries. Entries listed are controlled by the LIST parameters. You may list a single entry or all entries in the directory. Entries are in alphabetical order by network file name if NETFM is built with the SORT default on your operating system. If NOSRT5 option is selected or if SORT5 does not exist on the system, entries are not to be sorted in the output.

Format

LIST
PF=permanent file
NF=network file
UN=user name
PN=pack name
LO=list option

Parameters

PF=permanent file

The permanent file associated with the directory entry to be listed. If PF is specified, only the entry for the specified the permanent file is listed. Either PF or NF is required. If neither NF nor PF is specified, all entries are listed. If both PF and NF are specified, PF is ignored.

NF=network file

The CDCNET file name. If NF is specified, only the directory entry for the specified CDCNET file is listed; PF is ignored. Either NF or PF is required. For faster processing, use NF instead of PF. Wildcard characters are allowed in CDCNET file names on LIST.

UN=user name

User name under which a file is stored. If UN is specified, NETFM uses the user name specified when searching for the directory entry to list. If you omit UN, the user name in the directory entry is used. If you only specify the PF, the user name in the directory entry must match the user name under which you are running NETFM.

PN=pack name

Auxiliary device name. If PN is specified, only entries for files on the auxiliary pack named here are listed. If PN is 0 (zero), only entries for the files residing on the family device are listed. Default is to ignore the auxiliary device name when searching the directory.

LO=list option

List option desired. The allowed options are as follows:

Value	Description
F	Full listing of directory entries, including all pertinent information about the files from the directory entries.
blank (LO omitted)	Only CDCNET file names are listed.

Default is blank.

Examples This example lists the full directory entry for a CDCNET dump file, DUMP#FULL_08002510007F_850220103035.

```
LIST,NF=DUMP#FULL_08002510007F_850220103035,LO=F

DUMP#FULL_0  DF42AA  DIR  DL1  CXN0316 PACKNET  85/02/20.  85/02/20.
8002510007F        9               F            10.30.35.  10.30.35
_850220103035
```

PURGE (P)

Purpose Purges a NOS permanent file and deletes the file's CDCNET directory entry. You must be validated to purge the permanent file. If you try to purge a file without being validated, the directory entry is deleted.

Format

PURGE
PF=permanent file
NF=network file
UN=user name
PN=pack name
PW=password

Parameters

PF=permanent file

Permanent file to purge. If specified, the permanent file is purged and its directory entry is deleted. Either PF or NF is required. If both PF and NF are specified, PF is ignored. If only PF is specified, you must execute NETFM from the user name where the file is stored.

NF=network file

CDCNET file name. If NF is specified, the CDCNET file is purged and its directory entry is deleted. Either NF or PF is required. If NF and PF are specified, PF is ignored. For faster processing, use NF instead of PF. Wildcard characters are allowed in CDCNET file names on PURGE.

UN=user name

User name where the file is stored. If UN is specified, NETFM uses the specified user name to search for the file to purge. If you omit UN, the user name in the directory entry for the file is used when NF is specified, and the job's current user name is used if only PF is specified.

PN=pack name

Auxiliary device name. If PN is specified, the permanent file on the specified auxiliary pack is deleted. The pack name must match the pack name stored in the directory entry. If PN is 0 (zero), the permanent file residing on the system default family is deleted. Default is to ignore the pack name when searching the directory.

PW=password

Password required to purge the permanent file.

Examples This example purges CONFIGURATION#08002510007F and its directory entry. The response follows.

```
PURGE,NF=CONFIGURATION#08002510007F

PURGED  CONFIGURATION#0 C123456  DIR    NETOPS 84/03/22. 84/03/22.
        8002510007F             53           F 15.35.01. 17.50.15
```

UPDATE (U)

Purpose Updates the CDCNET directory to match the NOS permanent file base. If no parameters are specified, all directory entries are updated.

If the ABORT option has been specified on the NETFM command, and one or more matching entries cannot be updated (file either is missing or cannot have a CATLIST performed on it), NETFM aborts after updating all matching entries that can be updated.

Format

UPDATE
PF=permanent file
NF=network file
UN=user name
PN=pack name
C=comment
RT=retain timer

Parameters

PF=permanent file

The permanent file for which the directory entry is to be updated. If PF is specified, only the entry for the special permanent file is updated. If PF and NF are specified, PF is ignored. If only PF is specified, you must execute NETFM from the user name under which the file is actually stored.

NF=network file

The CDCNET file name. If NF is specified, only the directory entry for the specified CDCNET file is updated; PF is ignored. Use NF instead of PF for faster processing. Wildcard characters are allowed in CDCNET file names on UPDATE.

UN=user name

User name under which the file is stored. If UN is specified, NETFM uses the specified user name when searching for the file to be used for updating the directory entry. If you omit UN, the user name specified in the directory entry is used when NF is specified, and the job's current user name is used if only PF is specified.

PN=pack name

Auxiliary device name. If PN is specified, only entries for files on the specified auxiliary pack are updated. If PN=0, only entries for files on the family device are updated. Default is to ignore the auxiliary device name when searching the directory.

C=comment

Comments to be updated or added for the directory entry. Comments are specified as a string surrounded by quotation marks ("). If comments are specified, they replace comments currently stored in the directory entry. Comment strings can range from 1 through 50 6-bit display code characters. Additional characters are truncated. If the string contains a quotation mark, two consecutive quotation marks must be used to designate a quotation mark in the comment text.

RT=retain timer

Value for the retain flag in the directory entry, which determines the directory access and NOS permanent file attachment process. This parameter is not normally needed on UPDATE; it is meaningful only for files accessed by Network File Server (NETFS). Allowable keyword values are the following:

Value	Description
Y	Set retain flag.
N	Do not set retain flag.

Default is N.

If RT=Y, directory access and attachment of files are performed by NETFS only the first time inactive files are opened. Files remain attached by NETFS until one of the following conditions occurs:

- The network shuts down.
- NETFS terminates abnormally.
- The NETFS retain timer for the file expires.

See the CREATE description for default retain timer values for files supported by NETFS.

If RT=N, NETFS accesses the directory and attaches a permanent file each time it receives a request to open a file while a file is inactive. To reduce performance overhead and central memory requirements, specify RT=Y for files that NETFS frequently accesses, such as library files. RT=Y should be set for MODULE#.

Remarks If the permanent file for an entry does not exist, NOS issues a dayfile message indicating that the file was not found, along with the CDCNET file name and NOS permanent file name. Only entries for files on which you can perform CATLIST and ATTACH are updated. See the section on permanent file commands in the NOS Reference Set, Volume 3, for restrictions on the CATLIST command.

When executing UPDATE, NETFM internally gets or attaches the permanent files specified by PF or NF using local file ZZZZZFM. To prevent file-busy errors, return all direct access files in write mode to be accessed by UPDATE before accessing NETFM.

Examples This example changes the retain timer for object library file, MODULE#0635.

```
UPDATE,NF=MODULE#0635,RT=Y

UPDATED MODULE#0635  L700635  IND     ABC0123  84/08/01. 84/08/02.
        D1                 20             F   16.12.01. 16.22.45.
```

This example attempts to change the retain timer for terminal user procedure USER_PROCEDURE#1. However, since NETFM can't find a NOS permanent file for USER_PROCEDURE#1, the update does not succeed.

```
UPDATE,NF=USER_PROCEDURE#1,RT=Y

***NO PERM FILE OR NOT CATLISTABLE,NF=USER_PROCEDURE#1
```

This example attempts to update the directory entry for DI configuration procedure CONFIGURATION#08002510007C, but the entry for the file cannot be found.

```
U,NF=CONFIGURATION#08002510007C,PF=C10007C

***ENTRY NOT FOUND IN CDCNET DIRECTORY.
```

VERIFY (V)

Purpose

Verifies that a NOS permanent file exists for a CDCNET directory entry. If no parameters are specified, all directory file entries are verified.

You can use VERIFY directive after changing NOS permanent files. For example, you can check if any files have been inadvertently purged but not removed from the directory.

Only entries for files on which you can execute CATLIST or ATTACH are verified. See the section on permanent file commands in the NOS Reference Set, Volume 3, for restrictions on the CATLIST.

If the ABORT option has been specified on the NETFM command and if one or more matching entries cannot be verified (file either is missing or cannot have a CATLIST or ATTACH performed on it), NETFM aborts after verifying all of the verifiable matching entries.

Format

VERIFY
PF=permanent file
NF=network file
UN=user name
PN=pack name

Parameters

PF=permanent file

The permanent file for which the directory entry is to be verified. If PF is specified, only the entry for the specified permanent file is verified. If PF and NF are specified, PF is ignored.

NF=network file

The CDCNET file name. If NF is specified, only the directory entry for the specified CDCNET file is verified; PF is ignored. For faster processing, use NF instead of PF whenever possible. Wildcard characters are allowed in CDCNET file names on VERIFY.

UN=user name

User name under which the file is stored. If UN is specified, NETFM uses the specified user name when searching for the file to verify against its directory entry. If you omit UN, the user name in the directory entry is used when NF is specified, and the job's current user name is used if only PF is specified.

PN=pack name

Auxiliary device name. If PN is specified, only entries for files on the specified auxiliary pack are verified. If PN=0, only entries for files on the family device are verified. Default is to ignore the auxiliary device name when searching the directory.

Remarks When executing VERIFY, NETFM internally uses GET and ATTACH to access the permanent files specified by PF or NF (using local file ZZZZZFM). To prevent file-busy errors, return all direct access files in write mode to be accessed by VERIFY before accessing NETFM.

Examples This example verifies the directory entry for boot file called BOOT#0524_MCI.

```
VERIFY,NF=BOOT#0524_MCI

VERIFY GOOD, NF=BOOT#0524_MCI.
```

This example uses wildcard characters to specify that all directory entries for boot files should be verified. The VERIFY directive uses an asterisk (*) after BOOT#. Responses show whether each verification succeeds or fails.

```
V,NF=BOOT#*

VERIFY GOOD, NF=BOOT#0527_ESCI.

VERIFY GOOD, NF=BOOT#0527_MCI.

***VERIFY ERRORS.

***NO PERM FILE OR NOT CATLISTABLE,NF=BOOT#0529_MCI
```

Other NOS Network File Management Commands

Three additional utilities aid in NOS CDCNET directory and file management. These utilities are briefly described here. Messages generated by the utilities are listed in the NOS Operations handbook.

Utility	Description
NETBDF	Builds a directory file with file name NETDIR in the current user's permanent file catalog. NETBDF runs as part of the CDCNET installation process. NETBDF may also be used if a new directory must be built.
NETMDF	Merges directory files. Takes an existing directory and adds entries in that file to another directory. Checks for duplicate entries.
NETRDF	Restructures a directory file. Creates a new version of the directory so more entries can be added to it without degrading performance.

NETBDF (Build CDCNET Directory File)

Purpose Builds a CDCNET directory file named NETDIR in the catalog of the current user. Initializes directory for use by NETFM and NETFS. Optionally gives other users write permission on the directory. NETBDF runs automatically as part of CDCNET installation, and creates NETDIR on user name NETOPS. You can also use NETBDF to build a new CDCNET directory.

Format

NETBDF
NF=number of files
I=input file
delete option

Parameters *NF=number of files*

Total number of file entries the directory is expected to contain. NF may range from 0 through 10000. Default is 50 entries.

I=input file

Input file containing user names with write permission on the directory. The I parameter may have the following values:

Value	Description
I	User names with write permission are on file INPUT.
I=lfn1	User names with write permission are on local file lfn1 (file name of your choice).
I=0	No user names are to be granted write permission.
I (omitted)	User names with write permission are on file INPUT.

Default is I (omitted).

delete option

Delete option, specified by the keyword DELETE. If specified, the old CDCNET directory (if one exists) is deleted. If you don't specify DELETE and a directory already exists under this user name, NETBDF aborts with a dayfile message.

Remarks The input file specifying user names having write permission on the directory must be in display code, with one user name per line. Do not use characters such as periods or commas. At a minimum, the input file usually contains NETADMN.

After the directory has been created, you can add and delete user names having write permission on the directory using the PERMIT command.

Directory processing efficiency degrades if the directory contains more than 160 percent of the entries it was expected to contain.

Examples This example builds a CDCNET directory expected to have 45 entries.

```
NETBDF,NF=45,I=DIRUSER
```

Input file DIRUSER contains these user names permitted to update the directory:

```
NETADMN
INSTALL
```

NETMDF (Merge CDCNET Directory Files)

Purpose Merges CDCNET directory files. Moves directory entries from one directory file (source directory) to another directory file (destination directory). Checks files for duplicate entries. The user name you use to log in to NOS is the user name under which the destination directory is stored.

Format

NETMDF
UN=user name$_1$
F=permanent file
move option
replace option
SA=user name$_2$
secure option

Parameters **UN=user name$_1$**

User name under which the source directory resides.

F=permanent file

Permanent file name of the source directory to be merged with NETDIR. This file is in the catalog of the current user name (running the current job).

move option

Move option, specified by the keyword MOVE. If present, moves NOS permanent files associated with CDCNET file names in the source directory from their current location to the catalog of the user executing NETMDF. If omitted, no NOS permanent files are moved.

replace option

Replace option, specified by the keyword REPLACE. If present, replaces duplicate directory entries in the destination directory with entries from the source directory. If omitted and if matching directory entries exist, matching entries in the destination directory are not modified.

SA=user name$_2$

Site administrator's user name. If specified, and if files are moved by the MOVE option, the site administrator's user name is granted explicit write permission on any moved permanent files. Write permission is granted whether or not moved files have secure permanent file attributes (see secure option, next).

secure option

Secure option, specified by the keyword SECURE. If specified, files moved to the catalog of the destination directory's user name have secure permanent file attributes, regardless of attributes of original files. Secure permanent file attributes include the following:

CATEGORY=PRIVATE
MODE=READ
ALTERNATE CATLIST=NO

If move option is omitted, moved files have the same attributes as the original file, provided the original file's attributes include ALTERNATE CATLIST=YES. Otherwise, files have the secure file attributes shown above.

Remarks Use caution when several directory entries refer to the same permanent file name, even if they are under different user names.

Examples A network administrator at an interactive terminal uses NETBDF to build a source directory on NETADMN. Next, the administrator moves to the system console and logs in to NOS on user name NETOPS. The administrator enters NETMDF to merge the directory on NETADMN with the destination directory on NETOPS:

```
NETMDF UN=NETADMN.
```

NETRDF (Restructure CDCNET Directory File)

Purpose Restructures a CDCNET directory file. Creates a new version of the directory file that is structured more efficiently. The directory can grow to approximately 160 percent of the expected entries before performance begins to be degraded.

Format

NETRDF
NF=number of files

Parameters *NF=number of files*

Total number of file entries the directory file is expected to contain. Default is twice the number of entries currently in the existing CDCNET directory file.

Examples

```
NETRDF,NF=100.
```

Network Procedure Library Manager (NETPLM)

Network Procedure Library Manager (NETPLM) is a NOS utility that manages CDCNET procedure libraries for DI system configuration procedures, terminal definition procedures (TDPs), terminal user procedures (TUPs), and load procedures (LPs). NETPLM adds, deletes, and lists procedures in a procedure library.

Network File Manager (NETFM) is still used to manage exception lists and dump files, as well as individual configuration procedures before they are merged into a procedure library. In addition, NETPLM uses NETFM's routines to access the CDCNET directory and attach procedure library files. However, since most CDCNET configuration procedures are stored in procedure libraries on NOS, NETPLM is the main utility used to manage CDCNET configuration procedures.

NOTE

If your CDCNET directory contains entries for an individual configuration procedure and a configuration procedure library entry of the same name, Network File Server accesses the individual configuration procedure instead of the one in the procedure library.

There are procedures that help you use NETPLM: GETCP and REPCP. These procedures execute NETFM and NETPLM directives required for CDCNET configuration procedures and procedure libraries on NOS.

NETPLM Overview

When NETPLM executes, it does the following:

1. Parses and validates NETPLM command parameters.
2. Determines files to be accessed, and the manner in which they are accessed.
3. Validates the input files. Input files include the *addfile*, the file of procedures to be added. The addfile is specified by the A parameter on the NETPLM command. Input files also include the input file of NETPLM directives, specified by the I parameter on the NETPLM command. From these input files, NETPLM identifies required operations. During validation, NETPLM checks for correct procedure format, such as PROC and PROCEND statements. If input files are invalid, NETPLM aborts after identifying the errors.
4. If the operational CDCNET procedure library is to be modified, the procedure library is attached (by NETFM) in write mode through the network directory. The procedure library is then immediately reattached in read mode. This allows Network File Server (NETFS) to access the file, but locks out any other attempted NETPLM updates on the procedure library. The procedure library is copied into a temporary file. If an operational procedure library attached by NETFM is not to be modified, the operational library is returned.
5. Procedures in the addfile are added or replaced in the procedure library. If more than one procedure of the same name exists in the addfile, the last one in the addfile is added or replaced.
6. NETPLM directives are processed in the order they occur in the input file.

7. The temporary copy of the new procedure library is copied into the file specified on the NETPLM command. If the file to be updated is the operational CDCNET procedure library, it is reattached in write mode, and the temporary copy of the new library is copied into it. Then, the operational procedure library is returned or replaced. If the operational CDCNET procedure library is busy, the new procedure library is written to local file ZZZZZNL. NETPLM waits until the operational procedure library is no longer busy to copy the new library into the operational library, or until a user break 2 is entered (NETPLM aborts).

8. Temporary files are cleaned up and NETPLM terminates.

Library File Interlock

A library file interlock occurs only when NETPLM's NL and PL parameters specify that an operational procedure library is being updated. NETPLM initially attaches the operational procedure library file in write mode. If any user (including NETFS) has this file attached, a file-busy message is sent to the user, and NETPLM waits for the file to be available. Once the file is attached, NETPLM immediately changes the mode to read without releasing the file. No other NETPLM command can attach the file, because its write access request fails while the original NETPLM command keeps the file in read mode. NETPLM then creates a new procedure library on a local file. When the new library is created, NETPLM attempts to change its attach mode to write (if the file is busy, NETPLM informs the user and waits until the file is not busy). Once NETPLM gets the old library file in write mode, it copies the local file version of the new library onto the permanent file, then returns the file.

This library file interlock works only for direct access files. If you use indirect access files for your procedure libraries, NETPLM still runs correctly, but without the file interlock.

File Permissions and Constraints

To use NETPLM, you must have read permission to the CDCNET directory (NETDIR). You must also have read and write permissions for the operational procedure libraries to list or modify any of the procedures in the libraries.

Multiple NETPLM operations can run concurrently. However, a file interlock prevents procedures from being overwritten by allowing only one modification operation at a time per library (see Library File Interlock). For the interlock to work, use direct access rather than indirect access files for procedure libraries. If you use indirect access files, your files may be overwritten.

NETPLM Command

Purpose Accesses Network Procedure Library Manager (NETPLM), which allows you to add, delete, and list procedures on a CDCNET procedure library. The PL, NL, and UN parameters allow you to directly modify operational procedure libraries while your network is running. All file name parameters can specify local files. Parameters cannot be specified in positional format (that is, use the format: parameter_name=parameter_value).

Format

NETPLM
A=addfile
abort option
G=getfile
I=input file
L=list file
NL=new library
PL=previous library
PT=procedure type
replace option
UN=user name
z option

Parameters *A=addfile*

The local file containing new or replacement procedures to be used to update the procedure library (the addfile). You can specify a file name of your choice (A=filename), use A as a keyword, specify A=0, or omit A. Procedure adding or replacing (see following description of replace option) *always* occurs before any directives on the input file (see I parameter) are processed. If A is a keyword, the default is ADDFILE. If A is omitted or A=0, no procedures are added or replaced.

abort option

Abort option, specified by the keyword ABORT. This option specifies whether or not NETPLM should process directive warnings as errors. If ABORT is specified, a directive warning causes NETPLM to abort (after scanning directives for more errors). If ABORT is omitted, directive warnings do not cause NETPLM to abort.

G=getfile

The local file in which procedures requested on GET directives (see I parameter) are listed (the getfile). You can specify a file of your choice, use G as a keyword, specify G=0, or omit it. If G is a keyword, the default is GETFILE. If G is omitted or G=0, no procedures are listed.

I=input file

The name of a local file containing one or more NETPLM input directives. You can specify a file of your choice, use I as a keyword (specify I without a parameter value), specify I=0, or omit I. If I is omitted or specified as a keyword, the directives are read from file INPUT. If I=0, no directives are read. Directives are processed in order of their occurrence in the directive file. If the Z parameter is specified, its directives override any directives on the directive file.

The following NETPLM directives can be used in the directive input file:

NETPLM Directive	Description
DELETE procname D procname	Specifies a procedure to be deleted from the library.
GET procname or G procname	Specifies a procdure name (procname) with text to be written to the file specified on the G parameter. Text is from the current library, with all changes up to this directive. If you specify an asterisk (*) for procname, the text of all procedures in the current library is written in alphabetical order to the file specified on the G parameter.
LISTNAMES or LN	Specifies that the names of all the procedures in the current library are listed in alphabetical order to the file specified on the L parameter.
LISTPROC procname or LISTPROCS procname or LP procname	Specifies that the text of the procedure on this directive is listed in the file specified on the L parameter. If you specify an asterisk (*) for procname, the text of all procedures in the current library is listed in alphabetical order in the file specified on the L parameter.

L=list file

The local file to which NETPLM output, such as lists of procedure names or procedure listings, is written. This file includes a list of all changes or change errors for all procedures added, replaced, listed, and deleted.

You can specify a file of your choice, use L as a keyword (specify L without a parameter value), specify L=0, or omit L. If no file name is specified (L as a keyword value, L=0, or L omitted), the default is OUTPUT.

If a LISTNAMES directive is specified in a NETPLM session, the list file contains a list of all procedure names on the current library and all changes made up to the LISTNAMES directive.

If a LISTPROC directive is specified in a NETPLM session, the list file contains the text of procedures in the current library.

NL=new library

The local file to which the new library is written. You can specify a file of your choice, use NL as a keyword (specify NL without a parameter value), specify NL=0, or omit NL. If NL is a keyword, NETPLM attempts to update the actual procedure library using the new library. If NL=0, no new library is generated. If NL is omitted, the new library is written onto a local file named *xxxxLIB*, where *xxxx* is the procedure type specified on the PT parameter (TDP, TUP, SC, or LP). The resulting file names are TDPLIB, TUPLIB, SCLIB, and LPLIB, respectively. The NL file is rewound before and after NETPLM processing (procedure libraries are expected to be a single record).

PL=previous library

The local file containing the procedure library to be updated. You can specify a file of your choice, use PL as a keyword (specify PL without a parameter value), specify PL=0, or omit PL. If PL is specified as a keyword, or PL is omitted, NETPLM attempts to use the operational (NETDIR-specified) procedure library as the previous library. If PL=0, no previous library is used, and the new library only contains procedures from the addfile. The PL file is rewound before and after NETPLM processing (procedure libraries are expected to be a single record).

PT=procedure type

The type of procedure library to be updated. This parameter is required and has no default. The following values are allowed.

Keyword Value	Description
SC	DI system configuration procedure
TDP	Terminal definition procedure
TUP	Terminal user procedure
LP	Load procedure

replace option

Replace option, specified by the keyword R. R specifies whether or not replacing procedures in the library with procedures from the addfile is allowed. If R is specified, replacing existing procedures in the library with procedures from the addfile is allowed. If R is omitted, replacing existing procedures is not allowed.

UN=user name

The user name from which the network directory (NETDIR) is accessed. UN is ignored unless the operational library access option is selected by PL or NL. You can specify a user name of your choice, specify UN as a keyword (specify UN without a parameter value), or omit UN. If a value is specified for UN, the NETDIR in the specified user name is accessed. If UN is specified as a keyword, the NETDIR in the user name specified by NETFM's secondary default (NETOPS, in the released software) is accessed. If UN is omitted, the NETDIR in the current user name is accessed.

z option

Specifies whether or not the NETPLM command, rather than a separate input file, contains NETPLM directives. Specify this option by the keyword Z.

One or more NETPLM input directives (DELETE, GET, LISTNAMES, LISTPROC) are allowed on the NETPLM command. Z directives override directives on an input file.

If the Z option is used, the NETPLM command has the following structure:

NETPLM,$param_1$,$param_n$,Z./dir_1/dir_2../dir_n

where:

Parameter	Description
$param_1$,$param_n$	One or more NETPLM parameters.
period (.)	The NETPLM command terminator. Directives begin after the period.
slash (/)	The directive delimiter. This delimiter is the first character following the NETPLM command terminator. It can be any character except a blank, and should be used consistently within the command.
/dir_1/dir_2../dir_n	One or more NETPLM directives, separated by the directive delimiter.

Using the Z option, the NETPLM command and its directives must be 80 characters or less.

If Z is not specified on the NETPLM command, it is assumed that input directives are in an input file specified on the I parameter.

Responses NETPLM messages are documented in the NOS 2 Operations handbook.

Remarks Two or more of the A, G, and L parameters cannot specify the same file name.

Files (other than those specified by PL and NL) are not positioned before NETPLM processes them. After processing, files are positioned at the EOF or EOR following processed data.

Examples This example adds procedures on file NEWPROC to the operational TUP library on user name NETADMN. No directive is required. Output goes to file OUTPUT.

```
NETPLM,I=0,NL,A=NEWPROC,UN=NETADMN,PT=TUP
```

This example lists names of procedures, and text TEST1 procedure in the operational TUP library on file LIST. The Z parameter is used to add NETPLM directives after the NETPLM command. A semicolon is used as the directive delimiter.

```
NETPLM(PL,UN,PT=TUP,L=LIST,Z);LISTNAMES;LISTPROC TEST1
```

Library File Installation

After a procedure library is generated, it should have an entry in the CDCNET network directory, so it can be accessed by Network File Server (NETFS). If a directory entry does not exist for a procedure library on the network directory (NETDIR), you must add an entry using NETFM (see NETFM section of this appendix). The NOS CDCNET installation procedures install initial default procedure libraries and their directory entries.

When procedure libraries are processed as local files, it is your responsibility to copy the resulting new procedure library into the actual permanent file (as specified on NETDIR).

If no previous library (PL parameter) exists, you can specify PL=0 on the NETPLM command. NETPLM then generates a new library (NL parameter) which only contains procedures from the addfile. If no addfile is specified on the NETPLM command, the new procedure library NETPLM generates is empty.

File Formats

The following sections describe format requirements for files used with NETPLM.

Addfile (A Parameter)

The addfile is a local 6/12 ASCII file containing one or more new or modified procedures. This file is optional. If omitted on the NETPLM command, no procedures are added or replaced in the procedure library. The addfile can have any name.

The following format conventions apply to procedures specified in the addfile:

- Procedures must begin with a PROC statement and end with a PROCEND statement.
- Each procedure name must be 31 or fewer alphanumeric and special characters (special characters include #, _, @, and $).
- All procedure names should be unique in the procedure library. If a duplicate procedure name is added to the procedure library, the duplicate replaces the older procedure.
- Multiple blanks, commas, comments, and line continuations are allowed.
- All comment lines after the BOI or the previous procedure's PROCEND statement are considered part of the procedure.
- The PROCEND statement can optionally include the procedure's name, which must match the name specified on the PROC statement.

Input Directive File (I Parameter)

The input directive file is a local 6/12 ASCII file containing directives for processing the procedure library. This file is optional, and is used when deleting or listing procedures in the procedure library (using DELETE, GET, LISTNAMES, and LISTPROC directives).

The following format conventions apply to directive files:

- The directive file may contain one or more directives.
- One or more blanks can separate the directive command (such as D, G, LN, LP, DELETE, GET, LISTNAMES, LISTPROC, and LISTPROCS) from the procedure name.
- Only one directive is allowed per line.
- Blank lines are allowed.

NETPLM Interactive Directive Mode

You can enter NETPLM directives in interactive directive mode. To use this mode, specify a file assigned to an interactive terminal device as the input file (I parameter), and a file assigned to an interactive terminal for the output parameter (L parameter) (only possible when NETPLM is run from IAF). Normally, files INPUT and OUTPUT are assigned to the terminal for an interactive job. In interactive directive mode, directives are read from the keyboard. NETPLM sends a prompt (?) for each directive. As each directive is read and processed, output is sent to the terminal screen. NETPLM prompts for, and processes, more directives until you enter a carriage return without directives, signifying the end of directive input. In interactive directive mode, directive errors are not fatal. NETPLM identifies the error and prompts you to reenter a corrected directive.

List File (L Parameter)

If no fatal validation errors occur, the list file contains the following:

- A list of procedures added or replaced in the library (if any)
- A list of directives processed (if any)
- The resulting listings of any LISTPROC and LISTNAMES directives

If a fatal validation error occurs, the list file contains the following message:

```
FATAL ERROR FOUND
```

followed by two list sections:

- ADDFILE SUMMARY: Lists procedures in the addfile as valid or invalid, and indicates errors.
- INPUT DIRECTIVES: Lists all directives and indicates errors.

If a fatal error occurs, or a nonfatal error occurs and ABORT was specified on the NETPLM command, NETPLM terminates without updating the procedure library.

Format control characters are written into the file specified on the L parameter, unless the file is assigned to a terminal output device.

Getfile (G Parameter)

The text of one or more of the procedures on a procedure library can be put in the getfile (G parameter) using GET directives. The getfile contains procedures in the same format as those originally entered through the addfile. Once procedures are copied into the getfile, they can be modified and added back into the procedure library as an addfile during a later NETPLM run.

ZZZZZNL (Temporary New Library)

If the new procedure library cannot be written onto the CDCNET operational procedure library because the operational procedure library file is busy, local file ZZZZZNL contains the new procedure library when NETPLM aborts. You may then use this file to manually update the operational library.

CAUTION

Ensure that no other updates of the operational library have occurred; otherwise the update with new library on ZZZZZNL destroys other changes.

Example File Diagrams

Figures C-1 and C-2 show general formats of files read by NETPLM. While the examples use procedure type TUP, the same format is used for procedure types TDP, SC, and LP.

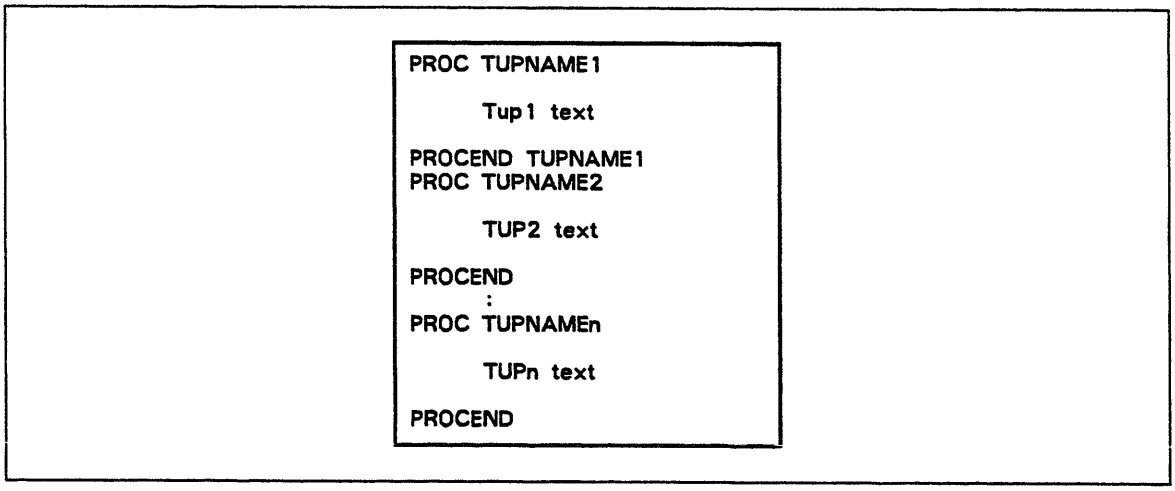


Figure C-1. Addfile Format Example

NOTE

All procedure names are unique, and may be up to 31 alphanumeric characters, plus special characters #, @, _, and $.

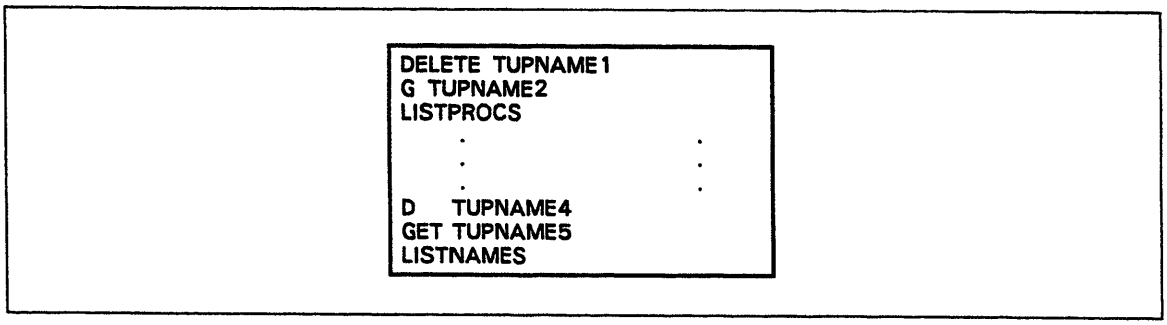


Figure C-2. Directive Input File Format Example

Manage CDCNET Configuration (MANCC) Utility D

This appendix shows how to access the Manage CDCNET Configuration (MANCC) utility. It also summarizes the MANCC subcommands. Complete instructions on using MANCC are in the CDCNET Configuration Guide.

Accessing MANCC

This section describes how to access MANCC on NOS/VE and NOS.

NOTE

Before using MANCC, check if your terminal supports a screen mode file editor. See the NOS/VE File Editor Tutorial/Usage or the NOS Full Screen Editor User's Guide for details on modifying EDIT_FILE or FSE at your terminal. If your terminal does not support screen mode EDIT_FILE or FSE, you must use MANCC in line mode.

Accessing MANCC on NOS/VE

To access MANCC on NOS/VE, do the following:

1. Make sure the appropriate terminal model has been specified by CHANGE_TERMINAL_ATTRIBUTES.
2. Enter:

```
CREATE_COMMAND_LIST_ENTRY ..
 ENTRY=$SYSTEM.CDCNET.VERSION_INDEPENDENT.COMMAND_LIBRARY
```

   This command adds MANCC commands to the set of commands you may enter. You might want to add this command to your user prolog.

3. Enter:

   MANAGE_CDCNET_CONFIGURATION or
   MANCC
   INPUT = file reference
   VERSION = 0..FFFF(16)
   STATUS = status variable

   INPUT (I)

   The input file from which MANCC reads subcommands. Default is $COMMAND.

   VERSION (V)

   The MANCC version to use for the session. The default version level is the value selected when MANCC is installed on the host. Each MANCC version generates and validates CDCNET configuration commands according to its corresponding CDCNET software version level. This parameter should be used only by CDCNET analysts who must create configuration files for DIs that are loaded with a CDCNET software version differing from the default.

   STATUS

   The status parameter here and on MANCC subcommands is the standard SCL status variable. For a description of this parameter, see the NOS/VE System Usage manual.

4. MANCC runs as a command utility that, except for the editing subcommand, operates in line mode. You can use DISPLAY_COMMAND_LIST_ENTRY (DISCLE) to see the list of MANCC subcommands, as in the following example.

```
/mancc
NC/discle
ENTRY  mancc
Commands

change_default                      create_configuration_file
display_configuration_file          edit_configuration_file
quit                                use_configuration_file
write_configuration_file            validate_configuration_file
NC/
```

Once you access MANCC, you can enter MANCC subcommands. The following prompt indicates when MANCC is ready to accept subcommands:

```
NC/
```

Accessing MANCC on NOS

To access MANCC on a NOS host, enter:

1. `SCREEN,terminal_type`

   If a screen definition is not in your user prolog, enter the SCREEN command to define the full screen interface for your terminal, for example, SCREEN,721.

2. `ASCII`

   The ASCII command allows MANCC messages to be displayed in a readable format. Also, certain characters that may be in configuration files, such as @ and :, have different encoding formats in ASCII and NORMAL modes.

3. `MANCC,I=filename,V=vvvv`

   I specifies the input file from which MANCC reads subcommands. The default for *I* depends on the job origin:

Job Origin	Default Input File
Interactive	Interactive terminal.
Batch	File INPUT (input is read from local file INPUT).

   If the specified input file is assigned to an interactive terminal, the MANCC main menu appears.

   vvvv is the version of MANCC to use (four hexadecimal digits). The default version level is the value selected when MANCC is installed on the host. Each MANCC version generates and validates CDCNET configuration commands according to its corresponding CDCNET software version level. This parameter should be used only by CDCNET analysts who must create configuration files for DIs loaded with a CDCNET software version differing from the default. To change the MANCC version level, see the SET_VERSION_LEVEL procedure in the NOS Installation handbook.

4. The MANCC main menu appears. This is the main menu where you select MANCC subcommands.

```
MANCC Version vvvv COPYRIGHT CONTROL DATA CORPORATION yyyy, yyyy

              1.  Create_Configuration_File   (CRECF)
              2.  Use_Configuration_File      (USECF)
              3.  Validate_Configuration_File (VALCF)
              4.  Display_Configuration_File  (DISCF)
              5.  Edit_Configuration_File     (EDICF)
              6.  Write_Configuration_File    (WRICF)
              7.  Change_Default              (CHAD)
              8.  Quit

          Select from the list above and press NEXT: __
```

NOTE

vvvv = software version number, *yyyy* = year

After accessing MANCC, you can select any MANCC subcommand. Most MANCC screens have a HELP function key active.

MANCC aborts if it encounters an error when processing subcommands in batch mode, or when processing subcommands from an input file not assigned to an interactive terminal.

MANCC Subcommands

MANCC subcommands are summarized in table D-1. They are described in detail in the CDCNET Configuration Guide.

Table D-1. MANCC Subcommand Summary

MANCC Subcommand	Description
CREATE_CONFIGURATION_FILE (CRECF)	Creates a basic configuration file of the specified type. CRECF prompts for for information on the equipment you want to configure. MANCC combines this information with values it provides in the generated file. The generated files are limited in choices and parameter values, and can be modified using EDICF. CRECF cannot be entered in batch mode.
USE_CONFIGURATION_FILE (USECF)	Specifies the file to use with subsequent MANCC subcommands, such as EDICF, DISCF, WRICF, or VALCF. Either USECF, CRECF, or VALCF should be specified at the start of a MANCC session to indicate the current file to use with subsequent MANCC subcommands.
VALIDATE_CONFIGURATION_FILE (VALCF)	Validates a configuration file for correct command syntax. The validated file can then be used with subsequent MANCC subcommands, such as EDICF, DISCF, or WRICF. You can also validate a configuration file during an EDIT_CONFIGURATION_FILE session using the VALDAT editing function.
DISPLAY_CONFIGURATION_FILE (DISCF)	Displays a configuration file with header information (date, time, and file type). Used for producing configuration reports. The displayed file can be copied, viewed, and printed using operating system utilities.
EDIT_CONFIGURATION_FILE (EDICF)	Invokes a file editor plus special features for adding and changing information in CDCNET configuration files, such as prompting for command parameters, syntax validation, and restricting the types of commands allowed in a file. EDICF cannot be entered in NOS batch mode.
WRITE_CONFIGURATION_FILE (WRICF)	Writes a created or edited file to another file.
CHANGE_DEFAULT (CHAD)	Changes the local file to which the output from a MANCC session is written.
QUIT	Ends a MANCC session.

Network Configuration Utility (NETCU) E

NOTE

Control Data no longer supports NETCU.

Character Set F

This appendix lists the ASCII character set, with conversions to decimal, hexadecimal and octal codes.

Table F-1. ASCII Character Set

OXCH	DNCOL	LNCOL	LNCOL	Mnemonic Name or Meaning
000	00	000	NUL	Null
001	01	001	SOH	Start of heading
002	02	002	STX	Start of text
003	03	003	ETX	End of text
004	04	004	EOT	End of transmission
005	05	005	ENQ	Enquiry
006	06	006	ACK	Acknowledge
007	07	007	BEL	Bell
008	08	010	BS	Backspace
009	09	011	HT	Horizontal tabulation
010	0A	012	LF	Line feed
011	0B	013	VT	Vertical tabulation
012	0C	014	FF	Form feed
013	0D	015	CR	Carriage return
014	0E	016	SO	Shift out
015	0F	017	SI	Shift in
016	10	020	DLE	Data link escape
017	11	021	DC1	Device control 1 (X-ON)
018	12	022	DC2	Device control 2
019	13	023	DC3	Device control 3 (X-OFF)
020	14	024	DC4	Device control 4
021	15 OvTEM	025	NAK	Negative acknowledge
022	17	026	SYN	Synchronous idle
023		027	ETB	End of transmission block
024	18	030	CAN	Cancel
025	19	031	EM	End of medium
026	1A	032	SUB	Substitute
027	1B	033	ESC	Escape
028	1C	034	FS	File separator
029	1D	035	GS	Group separator
030	1E	036	RS	Record separator
031	1F	037	US	Unit separator

(Continued)

Table F-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
032	20	040	SP	Space
033	21	041	!	Exclamation point
034	22	042	"	Quotation marks
035	23	043	#	Number sign
036	24	044	$	Dollar sign
037	25	045	%	Percent sign
038	26	046	&	Ampersand
039	27	047	'	Apostrophe
040	28	050	(	Opening parenthesis
041	29	051	)	Closing parenthesis
042	2A	052	*	Asterisk
043	2B	053	+	Plus
044	2C	054	,	Comma
045	2D	055	-	Hyphen
046	2E	056	.	Period
047	2F	057	/	Slant
048	30	060	0	Zero
049	31	061	1	One
050	32	062	2	Two
051	33	063	3	Three
052	34	064	4	Four
053	35	065	5	Five
054	36	066	6	Six
055	37	067	7	Seven
056	38	070	8	Eight
057	39	071	9	Nine
058	3A	072	:	Colon
059	3B	073	;	Semicolon
060	3C	074	<	Less than
061	3D	075	=	Equals
062	3E	076	>	Greater than
063	3F	077	?	Question mark
064	40	100	@	Commercial at
065	41	101	A	Uppercase A
066	42	102	B	Uppercase B
067	43	103	C	Uppercase C
068	44	104	D	Uppercase D
069	45	105	E	Uppercase E
070	46	106	F	Uppercase F
071	47	107	G	Uppercase G

(Continued)

Table F-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
072	48	110	H	Uppercase H
073	49	111	I	Uppercase I
074	4A	112	J	Uppercase J
075	4B	113	K	Uppercase K
076	4C	114	L	Uppercase L
077	4D	115	M	Uppercase M
078	4E	116	N	Uppercase N
079	4F	117	O	Uppercase O
080	50	120	P	Uppercase P
081	51	121	Q	Uppercase Q
082	52	122	R	Uppercase R
083	53	123	S	Uppercase S
084	54	124	T	Uppercase T
085	55	125	U	Uppercase U
086	56	126	V	Uppercase V
087	57	127	W	Uppercase W
088	58	130	X	Uppercase X
089	59	131	Y	Uppercase Y
090	5A	132	Z	Uppercase Z
091	5B	133	[	Opening bracket
092	5C	134	\	Reverse slant
093	5D	135	]	Closing bracket
094	5E	136	^	Circumflex
095	5F	137	_	Underline
096	60	140	`	Grave accent
097	61	141	a	Lowercase a
098	62	142	b	Lowercase b
099	63	143	c	Lowercase c
100	64	144	d	Lowercase d
101	65	145	e	Lowercase e
102	66	146	f	Lowercase f
103	67	147	g	Lowercase g
104	68	150	h	Lowercase h
105	69	151	i	Lowercase i
106	6A	152	j	Lowercase j
107	6B	153	k	Lowercase k
108	6C	154	l	Lowercase l
109	6D	155	m	Lowercase m
110	6E	156	n	Lowercase n
111	6F	157	o	Lowercase o

(Continued)

Table F-1. ASCII Character Set *(Continued)*

Decimal Code	Hexadecimal Code	Octal Code	Graphic or Mnemonic	Name or Meaning
112	70	160	p	Lowercase p
113	71	161	q	Lowercase q
114	72	162	r	Lowercase r
115	73	163	s	Lowercase s
116	74	164	t	Lowercase t
117	75	165	u	Lowercase u
118	76	166	v	Lowercase v
119	77	167	w	Lowercase w
120	78	170	x	Lowercase x
121	79	171	y	Lowercase y
122	7A	172	z	Lowercase z
123	7B	173	{	Opening brace
124	7C	174	|	Vertical line
125	7D	175	}	Closing brace
126	7E	176	~	Tilde
127	7F	177	DEL	Delete

DI Reset Codes G

This appendix lists the DI reset codes numerically and suggests actions based on them. Table G-1 provides the numerical list of reset codes. The Action Code column in table G-1 is keyed to the suggested actions, which follow the table.

Table G-1. Numerical List of DI Reset Codes

Numeric Code	Reason Code	Issuing Component	Action Code
00(16)	power_up_reset	MPB ROM	DA
02(16)	manual_reset	MPB ROM	DA
03(16)	halt_memory_fault	MPB ROM	HW
04(16)	dead_man_time_out	MPB ROM	SW
05(16)	pp_channel_master_clear	ICA Boot	DA
06(16)	reset_function	ICA Boot	DA
10(16)	load_software_too_big	Initialization Bootstrap	LF
11(16)	improper_first_module	Initialization Bootstrap	LF
12(16)	unsatisfied_external	Initial Loader	LF
13(16)	sysconfig_not_loaded	Initial Loader	LF
14(16)	post_load_routines_not_found	Initial Loader	LF
15(16)	reset_at_end_of_quiesce	Initialization Bootstrap	DA
16(16)	unrecognizable_object_text	Initial Loader	LF
17(16)	duplicate_entry_point	Initial Loader	LF
18(16)	task_error_no_recovery_proc	System Ancestor	SW
19(16)	task_error_exceed_max_recovers	System Ancestor	SW
1a(16)	task_error_unrecoverable	System Ancestor	SW
1b(16)	no_configuration_file_obtained	Configuration Procurer	OP
1c(16)	configuration_file_read_error	Configuration Procurer	OP
1d(16)	not_enough_memory_for_buffers	Loader	LF
1e(16)	identification_record_expected	Loader	LF
1f(16)	unexpected_idr_encountered	Loader	LF
20(16)	premature_eof_on_file	Loader	LF
21(16)	absolute_length_too_large	Loader	LF
22(16)	invalid_object_text_version	Loader	LF
23(16)	invalid_module_kind	Loader	LF
24(16)	invalid_module_attribute	Loader	LF
25(16)	invalid_section_ordinal	Loader	LF
26(16)	duplicate_section	Loader	LF
27(16)	invalid_section_kind	Loader	LF
28(16)	invalid_allocation_alignment	Loader	LF
29(16)	invalid_offset	Loader	LF
2a(16)	storage_allocation_failed	Loader	OP/LF
2b(16)	undefined_section	Loader	LF
2c(16)	reference_outside_of_section	Loader	LF
2d(16)	invalid_address_kind	Loader	LF
2e(16)	invalid_number_of_bytes_spanned	Loader	LF
2f(16)	transfer_sym_entry_pt_not_found	Loader	LF
30(16)	parameter_verification_error	Loader	LF
31(16)	loader_table_not_found	Loader	LF

(Continued)

Table G-1. Numerical List of DI Reset Codes *(Continued)*

Numeric Code	Reason Code	Issuing Component	Action Code
32(16)	kill_system_with_dump	KILS Command	DA
33(16)	kill_system_without_dump	KILS Command	DA
34(16)	stop_executive	Executive	SW
35(16)	module_checksum_is_invalid	System Audits	LF
36(16)	software_dead_stop	DEAD STOP	SW
37(16)	fatal_parity_error	Executive	HW
38(16)	ac_low_error	Executive	OP
39(16)	temperature_shutdown_error	Executive	OP
3A(16)	reset_from_debugger	Hardwired in Debugger	DA
3B(16)	overflowed_stack	Exec/System Audits	SW
3C(16)	system_data_not_found	Initial Loader	LF
3D(16)	boot_file_media_mismatch	Boot Start-up Code	OP/LF
3E(16)	cybil_detected_error	CYBIL Routines	SW
3F(16)	hard_failure	Executive	HW/SW
40(16)	well_known_configuration_change	Configuration Procurer	NA
41(16)	mpb_ram_ptr_not_found	Initial Loader	LF
42(16)	timer_task_module_missing	Initial Loader	LF
43(16)	task_received_unknown_itm	Any Task	SW
44(16)	sna-3270_tip_dhcf_abort	SNA 3270 TIP_DHCF	SW
45(16)	configuration_cmd_read_error	Configuration Procurer	OP
46(16)	eeprom_updated	Configuration Procurer	NA
47(16)	loader_bus_error	Initial Loader	LF

Suggested Actions Based on DI Reset Codes

The remainder of this appendix describes the circumstances in which DIs reset and suggests actions to be taken based on various DI reset codes. This information is keyed to the Action Code column in table G-1 through the abbreviation given for each reset title. Reset code descriptions are organized numerically within the action code groups.

Some of the actions suggested here require tools or facilities that might not be available at your site. If you need further assistance, submit a programming system report (PSR) to Control Data.

Deliberate Action (DA)

These resets are due to human intervention. For resets that generate dumps, the following steps can be taken to obtain more information:

- Display the executive error table with the Dump Analyzer DISEET subcommand.
- Use the DISSCT subcommand to check for memory and/or buffer regulation.
- Display calls for the running task.
- Use the DISC subcommand to find any task calling DEAD_STOP, RESET_DI, or ABORT_SYSTEM.

Following are descriptions of the causes and suggested actions for the DI resets classified as deliberate actions:

00(16) = POWER_UP_RESET

No dump file is written under this condition.

02(16) = MANUAL_RESET

The toggle switch on the MPB was manually reset. Additional information should be obtained from the person who reset the system.

05(16) = PP_CHANNEL_MASTER_CLEAR

The ICA-II is reset during the host deadstart.

06(16) = RESET_FUNCTION

The ICA-II is reset via a reset function from the PP.

15(16) = RESET_AT_END_OF_QUIESCE

This occurs if a DI is manually reset while the onboard diagnostics are running, or if there was a channel error. If the host error log indicates a channel error, follow the hardware error reporting process.

32(16) = KILL_SYSTEM_WITH_DUMP

The system was reset by the KILL_SYSTEM operator command. Additional information should be obtained from the person who reset the system.

33(16) = KILL_SYSTEM_WITHOUT_DUMP

The system was reset by the KILL_SYSTEM operator command. Additional information should be obtained from the person who reset the system. No dump file is written under this condition.

3A(16) = RESET_FROM_DEBUGGER

The RS command was entered from the DI Resident Debugger. Additional information should be obtained from the person who reset the system.

No Action (NA)

This type of reset does not require any human intervention.

40(16) = WELL_KNOWN_CONFIGURATION_CHANGE

The system was reset to immediately and automatically force the changes specified in the configuration file for the MPB RAM.

There are a number of configurable values (such as which protocol stacks are enabled and data buffer size) that are associated with this reset. In addition, if a change in configuration affects the allocation of PMM, reset 40 is invoked. In any case, a dump is never taken when a reset 40 occurs.

46(16) = EEPROM_UPDATED

The system was reset to immediately and automatically force the changes first installed on the MPB-II board and/or one or more SMM4 boards in the DI. A dump is never taken when a reset 46 occurs.

Operational (OP)

The probable cause for each of these resets is something that can most likely be corrected on-site in the software or environmental conditions. The following suggested actions should be taken.

1B(16) = NO_CONFIGURATION_FILE_OBTAINED

Verify proper DI SYSTEM_ID at location 8400(16) by putting the DI in maintenance mode and using the DI console (see the CDCNET Hardware Installation and Troubleshooting manual). If your network is operating under NOS/VE, issue the ACTIVATE_NETWORK_FILE_ACCESS command. If there is no configuration file for the CDCNET system (or if it is busy or otherwise unavailable), an error is reported on the NOS/VE system job log display. Also, inspect the OCU library for a configuration file with the appropriate system identifier.

Under NOS, verify that NETFS is running properly by examining the NAM K-display. Also, use NETFM to list or attach the configuration file (using the NF parameter).

If no configuration file exists, create one. See the CDCNET Configuration Guide.

The CDCNET log message 20 = CFP_FILE_OPEN_ERROR also indicates that the configuration file could not be opened. The Configuration File Procurer tries three times to open the file (with a delay between tries). You can enter the BYPASS_CONFIGURATION (BYPC) command and enter the configuration commands manually. If the file is not accessible after three tries and no BYPASS_CONFIGURATION command has been received, the DI is reset.

1C(16) = CONFIGURATION_FILE_READ_ERROR

A configuration file read error occurred or the host file server became unavailable. Check the configuration file and check the status of the file server. If your network is operating under NOS/VE, examine the system job log display to determine if Network File Access restarted or terminated abnormally.

Under NOS, verify that NETFS is running properly by examining the NAM K-display.

The error may be due to using DIDEBUG while executing the configuration file commands, causing a loss of the logical link.

2A(16) = STORAGE_ALLOCATION_FAILED

This indicates that not enough memory was available when the Initial Loader was building the loader data structures for a module. Add more memory or remove modules from the boot file before reloading. This reset code is also listed under Load File (LF) action.

38(16) = AC_LOW_ERROR,
39(16) = TEMPERATURE_SHUTDOWN_ERROR

Environmental problems are suspect. Contact installation management personnel or customer engineers.

3D(16) = BOOT_FILE_MEDIA_MISMATCH

The boot file type loaded in the DI did not match the medium it was loaded across; for example, a channel boot file was loaded over ESCI instead of a channel. Look at field boot_map_entry_address in MPB RAM to find out what medium the DI was loaded across. See also this reset under the Load File (LF) heading.

41(16) = MPB_RAM_PTR_NOT_FOUND

The system MPB_RAM_PTR entry point (in EXEC_MPB or ICA_EXEC_MPB) is missing from the boot file. Rebuild the boot file, adding this module, before reloading.

42(16) = TIMER_TASK_MODULE

The TIMER_TASK_MODULE (EXEC_PMM or ICA_EXEC_MPB) is missing from the boot file. Rebuild the boot file, adding this module, before loading.

45(16) = CONFIGURATION_CMD_READ_ERROR

A configuration command read error occurred or the host file server became unavailable while a configuration file command was executing. Check the configuration file and check the status of the file server. Additionally, the log messages in the dump file should show which command this error occurred on. If your network is operating under NOS/VE, examine the system job log display to determine if Network File Access restarted or terminated normally.

Under NOS, verify that NETFS is running properly by examining the NAM K-display.

The error may be due to using DIDEBUG while executing the configuration file commands, causing a loss of logical link.

Load File (LF)

The probable cause for each of these resets is a bad load file. If the load file has never been used successfully before, get a correct file. If this load file has been used successfully before, a software or hardware problem is likely. The following descriptions assume the latter to be true.

10(16) = LOAD_SOFTWARE_TOO_BIG

The boot file is too large to fit into SMM. Remove unnecessary modules from the boot file library or add more SMM before reloading.

This reset may also indicate that the on-board diagnostics detected an SMM failure and have marked a block of SMM as unavailable. The remaining SMM is not sufficient for loading the boot file.

11(16) = IMPROPER_FIRST_MODULE

The first module in the boot file was not the Initial Loader (INITLDRABS). Check the boot file for irregularities, or to see if the library file might have been moved into a boot file by mistake.

12(16) = UNSATISFIED_EXTERNAL

The initial load failed because an entry point was referenced that was not externally declared by any module in the boot file. Missing entry point names are displayed on the DI console. Do a test link (using SES procedure) of the boot file object library after deleting any ABS modules.

13(16) = SYSCONFIG_NOT_LOADED

The SYS_CNFG table (in EXEC_MPB) is missing from the boot file.

14(16) = POST_LOAD_ROUTINES_NOT_FOUND

The entry INITIALIZE_EXECUTIVE (in POST_LOADER_PROCESSING) is missing from the boot file.

16(16) = UNRECOGNIZABLE_OBJECT_TEXT,
1E(16) = IDENTIFICATION_RECORD_EXPECTED,
1F(16) = UNEXPECTED_IDR_ENCOUNTERED,
20(16) = PREMATURE_EOF_ON_FILE,
21(16) = ABSOLUTE_LENGTH_TOO_LARGE,
22(16) = INVALID_OBJECT_TEXT_VERSION,
23(16) = INVALID_MODULE_KIND,
24(16) = INVALID_MODULE_ATTRIBUTE,
25(16) = INVALID_SECTION_ORDINAL,
26(16) = DUPLICATE_SECTION,
27(16) = INVALID_SECTION_KIND,
28(16) = INVALID_ALLOCATION_ALIGNMENT,
29(16) = INVALID_OFFSET,
2B(16) = UNDEFINED_SECTION,
2C(16) = REFERENCE_OUTSIDE_OF_SECTION,
2D(16) = INVALID_ADDRESS_KIND,
2E(16) = INVALID_NUMBER_OF_BYTES_SPANNED,
2F(16) = TRANSFER_SYM_ENTRY_PT_NOT_FOUND

Unknown or unsupported loader text records were found in the boot file or loader library. Check the file module library for irregularities. Check whether newly added modules were compiled with DIDEBUG on or by the wrong compiler or assembler.

17(16) = DUPLICATE_ENTRY_POINT

A duplicate entry point was detected. Do a test link (using SES procedure) of the boot file module library after deleting any ABS modules.

1D(16) = NOT_ENOUGH_MEMORY_FOR_BUFFERS

There must be enough memory after the initial load for allocation of 100 descriptor buffers and 65535 bytes of data buffers. If not, this reset code is issued. Remove unnecessary modules from boot file library before reloading.

2A(16) = STORAGE_ALLOCATION_FAILED

This indicates that not enough memory was available when the Initial Loader was building the loader data structures for a module. Add more memory or remove modules from the boot file before reloading. This reset code is also listed under Operational (OP) action.

30(16) = PARAMETER_VERIFICATION_ERROR

A compilation-time error was detected: the named procedure XDCL and XREF parameters don't match either in type or number. Types must match exactly; they may not match by synonyms or aliases. Assembler entry points must precede CYBIL references when the CYBIL references do not agree.

31(16) = LOADER_TABLE_NOT_FOUND

A crucial loader data structure was not found. This structure is in module OLL_PROGRAM_INTERFACE_PROCS. Under NOS/VE, use the DISPLAY_OBJECT_LIBRARY command to examine the boot file.

Under NOS, do a test link (using the SES.LINK68K procedure) on the object file to determine whether this module is in the boot file.

If the module is missing, add it before reloading.

35(16) = MODULE_CHECKSUM_IS_INVALID

SYSTEM_AUDITS aborted the system because a loaded module was corrupted. Use the Dump Analyzer to examine the SYSTEM_AUDITS stack for the module name and section ordinal, then use DISM to examine the affected memory for recognizable patterns. To locate the SYSTEM_AUDITS stack:

- Use DISTCB, with TI=ALL.
- Examine the output for task name SYSTEM_AUDITS.
- Display stack length number of bytes from the stack segment address in the SYSTEM_AUDITS TCB. This is the SYSTEM_AUDITS stack.

3C(16) = SYSTEM_DATA_NOT_FOUND

The SYSTEM_DATA entry point (in SYSTEM_AUDITS) is missing from the boot file. Rebuild the boot file, adding this module before reloading.

3D(16) = BOOT_FILE_MEDIA_MISMATCH

The boot file type loaded in the DI did not match the medium it was loaded across; for example, a channel boot file was loaded over ESCI instead of a channel. Look at field boot_map_entry_address in MPB RAM to find out what medium the DI was loaded across. Look at the XDCL'ed variable abort_message to see what the boot file type is.

47(16) = LOADER_BUS_ERROR

A bus error occurred during the initial load sequence. If any SMM errors occurred while leaving on-board diagnostics, the fault LED remains lit on the respective board(s).

Hardware (HW)

A hardware problem is the probable cause for each of these resets. Perform the suggested hardware problem isolation or correction.

03(16) = HALT_MEMORY_FAULT

A double-bit SMM error occurred. Check NPA reports to identify failing SMM board.

Board failure might show on indicator light if onboard diagnostics failed; this is seen after the DI resets and is going through diagnostics. See the CDCNET Hardware Installation and Troubleshooting manual.

37(16) = SMM_DOUBLE_BIT_ERROR

A double-bit SMM error occurred. Try the reset or power-on diagnostics to isolate the failing SMM board. See the CDCNET Hardware Installation and Troubleshooting manual.

3F(16) = HARD_FAILURE

This reset code is part of CDCNET's failure management feature. A hard failure is defined as one from which recovery is not possible; it can be caused by hardware or software.

In the error log file, this message indicates the board slot number for the DI subsystem where the failure occurred. A separate log message is issued for the failing subsystem or its failed software. Examine the executive error table for clues.

Software (SW)

A software bug is the probable cause for each of these resets. Submit a PSR with the dump file and CDCNET log file.

04(16) = DEAD_MAN_TIME_OUT

A running task took too long to execute, preventing SYSTEM_AUDITS from resetting the timer.

Use the Dump Analyzer to determine why the task timed out:

1. Use the DISTCB subcommand to identify the task with the task state RUNNING—this is the task that timed out.
2. Use the DISSCT subcommand to look at the following values:

   INTERRUPT FIREWALL CHAIN ADDRESS. This identifies the interrupt processor.

   BINARY TIME-OF-DAY. This indicates the millisecond clock value at the time of failure.

3. Use the DISM subcommand to look at the LAST_DEADMAN_RESET value in the system data record.

If the difference between the binary time-of-day and LAST_DEADMAN_RESET is less than 10,000(10) and the interrupt firewall chain address was 0, then the error is probably hardware-related. Try to isolate the problem using the CDCNET Hardware Installation and Troubleshooting manual before writing a PSR.

18(16) = TASK_ERROR_NO_RECOVERY_PROC,
19(16) = TASK_ERROR_EXCEED_MAX_RECOVERS,
1A(16) = TASK_ERROR_UNRECOVERABLE,
36(16) = SOFTWARE_DEAD_STOP

The task that caused the reset has a task state of RUNNING or SUSPEND.

34(16) = STOP_EXECUTIVE

Use the Dump Analyzer DISEET or DISM subcommands to examine the executive error table for error information. The field STOP_SUPERVISOR_STACK_POINTER contains the supervisor stack pointer at the time of the reset. Using DISM, display this stack. The top of the stack contains the return address to the caller of STOP_EXEC.

3B(16) = OVERFLOWED_STACK

A stack overflow was detected on a call to the Executive when the value of register A7 was numerically less than the first byte address of the stack, or by SYSTEM_AUDITS after a task had stopped. Use the VALSA subcommand to reveal violations of stack areas. Or, use DISC to check the affected task for recursive calling. Move large variables off the stack or increase the stack size. Use ALLOCATE/FREE rather than PUSH CYBIL statements, if feasible.

3E(16) = CYBIL_DETECTED_ERROR

CYBIL run-time routines detected an error (when compiled with range checking on). Correct code and rebuild the boot file before reloading.

3F(16) = HARD_FAILURE

This reset code is part of CDCNET's failure management feature. A hard failure is defined as one from which recovery is not possible; it can be caused by hardware or software. See the description of this reset code under the Hardware (HW) heading.

43(16) = TASK_RECEIVED_UNKNOWN_ITM

This reset code indicates that an unknown ITM task message was received by a task.

44(16) = SNA_3270_TIP_DHCF_ABORT

This reset code occurs when the TIP software traps an invalid request response unit.

Command Index