DECnet SNA
Gateway for Channel Transport
Gateway for Synchronous Transport

Management (OpenVMS)

Part Number: AA-LU43E-TE

November 1993

This document describes how to use the DECnet SNA Gateway management software to configure, monitor, and control Gateway systems.

Revision/Update Information: This is a revised manual.

Operating System and Version: OpenVMS V5.4 or later
OpenVMS AXP V1.5

Software Version: DECnet SNA Gateway-CT V2.1 or
Gateway-ST V1.2

November 1993

Contents

3 Gateway Management Utilities

4 Line Management

5 Circuit Management

6 PU Management

7 LU Management

8 Access Name Management

9 Server Management

10 Event Logging

11 Using NETTRACE

12 Using SNATRACE

13 SNANCP Commands

14 NCP Commands

D Event Messages

Index

Examples

Figures

Tables

Preface

The DECnet SNA Gateway-CT and Gateway-ST systems enable communication and sharing of information between Digital Equipment Corporation and International Business Machines Corporation systems. DECnet SNA access routines, such as the DECnet SNA 3270 Terminal Emulator for OpenVMS (3270 TE), and the DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS, are software products that use the Gateways to access IBM® applications and programs.

This manual provides information on how to configure, monitor, and control a Gateway system. Use this guide in conjunction with other Digital and IBM documentation.

Intended Audience

This manual is designed to assist system and network managers in managing Gateway systems. You must be familiar with the Digital and IBM systems you will be using.

Document Structure

This document contains the following chapters and appendixes:

- Chapter 1, Overview, describes the software, hardware, and components of the DECnet SNA Gateway-ST and the DECnet SNA Gateway-CT.
- Chapter 2, Gateway Security, provides an overview of the types of access control used to secure the Gateways.
- Chapter 3, Gateway Management Utilities, introduces the Gateway management utilities. This chapter describes the function of each utility, and tells you how to start, enter commands, and exit from each utility.
- Chapter 4, Line Management, explains how to manage SDLC and channel-attached lines.
- Chapter 5, Circuit Management, describes how to manage the SDLC and channel-attached circuits.

- Chapter 6, PU Management, explains how to manage the PUs for the Gateways.
- Chapter 7, LU Management, describes how to manage LUs for the Gateway.
- Chapter 8, Access Name Management, explains how to manage access names for the Gateways.
- Chapter 9, Server Management, describes how to manage the servers in the Gateways.
- Chapter 10, Event Logging, describes SNA event logging.
- Chapter 11, Using NETTRACE, explains how to use the NETTRACE utility to trace data traffic between a gateway and the SNA network.
- Chapter 12, Using SNATRACE, describes how to use the SNATRACE utility to trace data traffic between a gateway and the SNA network.
- Chapter 13, SNANCP Commands, contains a dictionary of all SNANCP commands.
- Chapter 14, NCP Commands, contains a dictionary of all NCP commands used to manage the DECnet connection between a load host or management node and a gateway.
- Appendix A, EBCDIC/DMCS Translation Tables, explains modifying and building translation tables.
- Appendix B, Counter Summary, describes the counters associated with the Gateway.
- Appendix C, Gateway Management Utility Messages, lists the messages that SNANCP, SNAP, NETTRACE and SNATRACE produce.
- Appendix D, Event Messages, lists the types of event messages logged.

A postpaid Reader's Comments form is located at the back of this manual. If you have any comments about this manual, please complete and return this form.

Associated Documents

The documentation for a DECnet SNA Gateway-ST system consists of the following sets of manuals:

- Access routine manuals
- DEC MicroServer manuals

- Gateway-ST software manuals

The documentation for a DECnet SNA Gateway-CT system consists of the following sets of manuals:

- Access routine manuals
- DEC ChannelServer and DEC ChannelServer II manuals
- Gateway-CT software manuals

DEC MicroServer Manuals

The following manuals document the DEC MicroServer:

- *Installing the DEC MicroServer*
- *DEC MicroServer Systems Configuration Card*

The following manuals document the DEC MicroServer-SP:

- *Installing the DEC MicroServer-SP*
- *DEC MicroServer Systems Configuration Card*

DECnet SNA Gateway-ST Software Manuals

The following manuals are part of the DECnet SNA Gateway-ST documentation kit:

- *DECnet SNA Gateway for Synchronous Transport Installation*
- *DECnet SNA Gateway for Synchronous Transport and Gateway for Channel Transport Management*
- *DECnet SNA Gateway for Synchronous Transport Problem Solving*
- *DECnet SNA Gateway for Synchronous Transport Guide to IBM Parameters*

DEC ChannelServer Manuals

The following manuals document the DEC ChannelServer II:

- *DEC ChannelServer II Installation*
- *DEC ChannelServer II Problem Solving*
- *DEC ChannelServer II Identification Card*

The following manuals document the DEC ChannelServer:

- *DEC ChannelServer Hardware Installation Guide*
- *DEC ChannelServer Hardware Troubleshooting Guide*
- *DEC ChannelServer Identification Card*

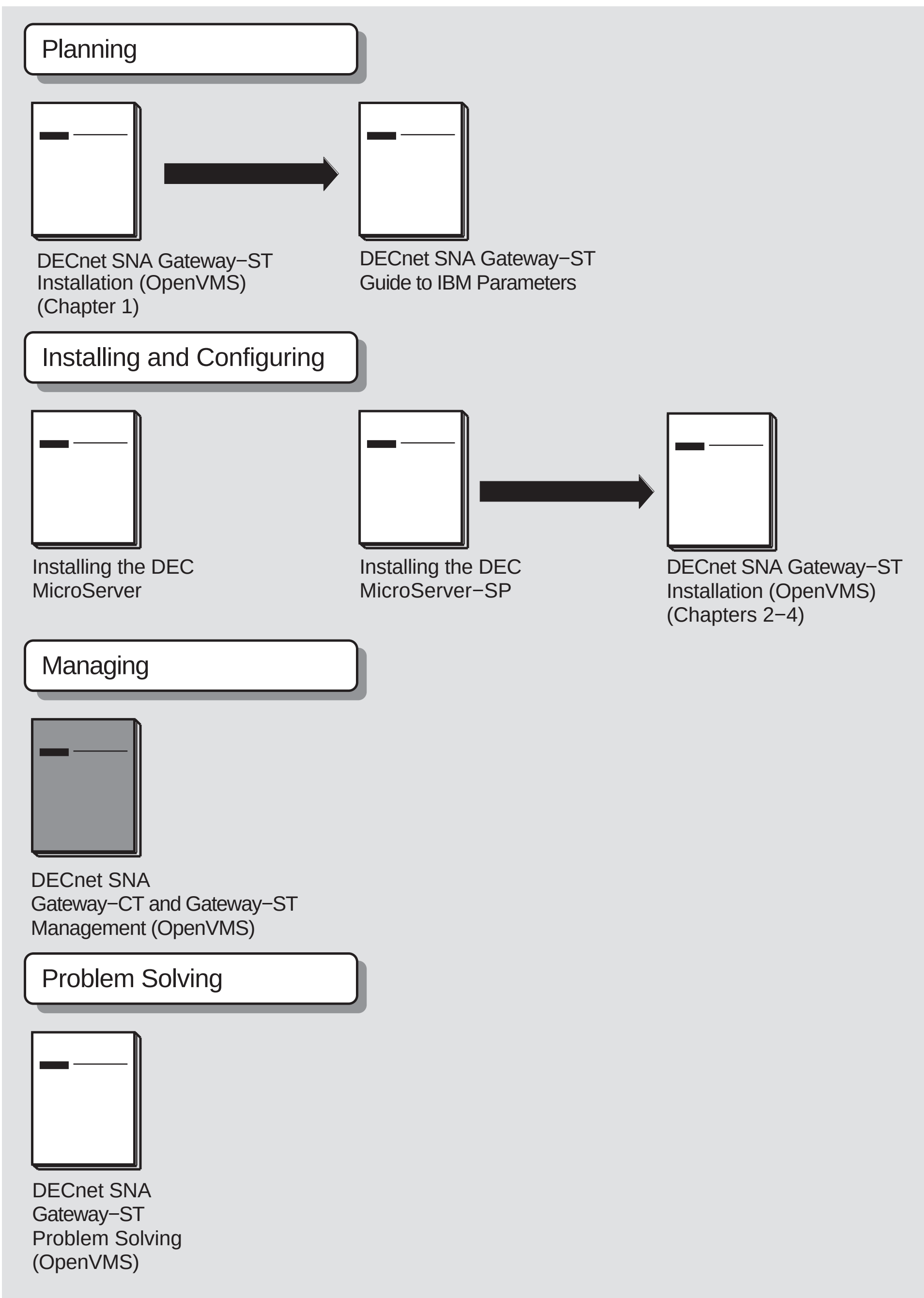
DECnet SNA Gateway-CT Software Manuals

The following manuals are part of the DECnet SNA Gateway-CT documentation kit:

- *DECnet SNA Gateway for Channel Transport Installation*
- *DECnet SNA Gateway for Synchronous Transport and for Channel Transport Management*
- *DECnet SNA Gateway for Channel Transport Problem Solving* (OpenVMS and ULTRIX)
- *DECnet SNA Gateway for Channel Transport Guide to IBM Parameters*

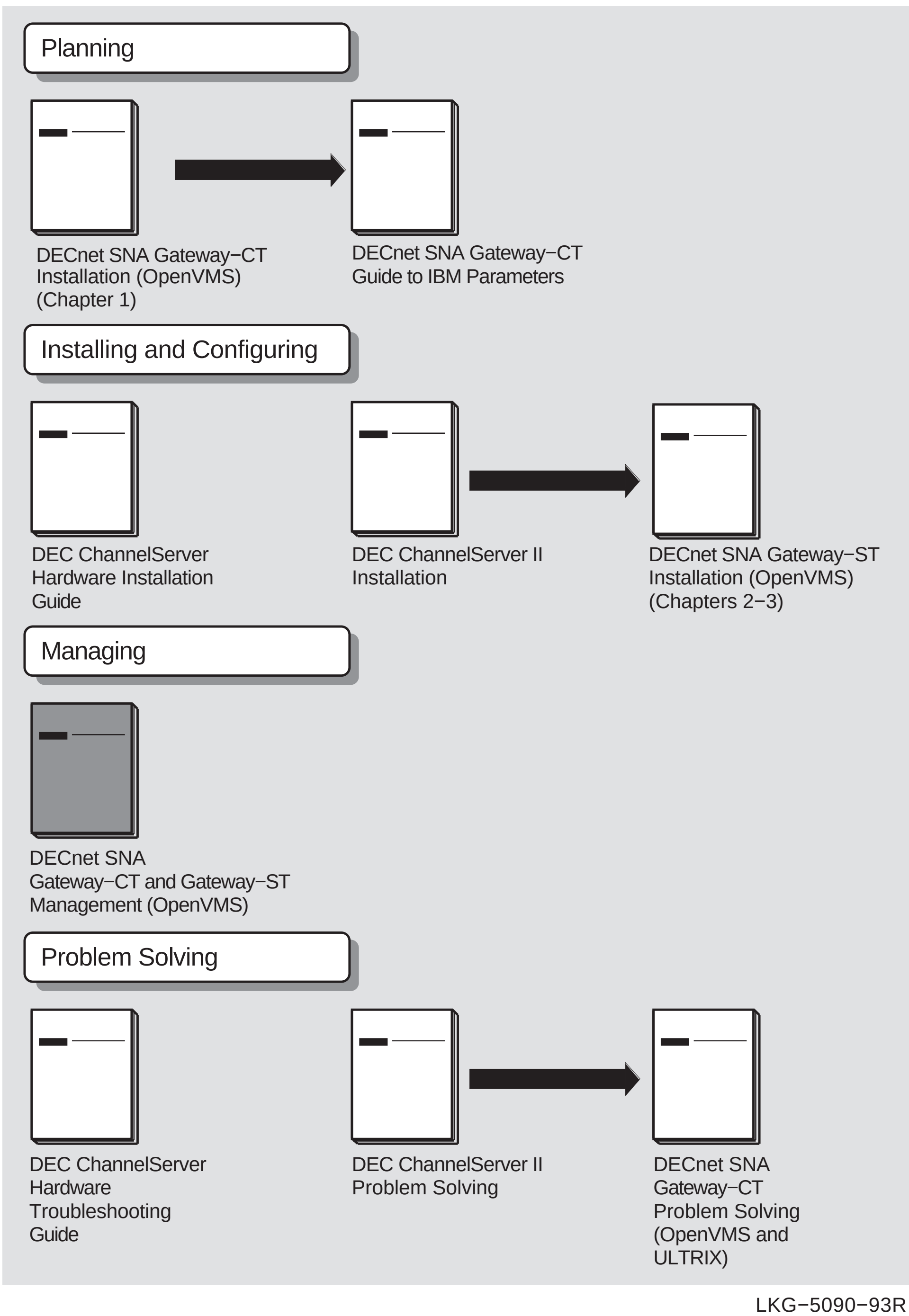
Figure 1 and Figure 2 list the Gateway-ST and Gateway-CT manuals according to the major tasks they describe. The arrows in the figure indicate the logical order in which to use the books. The highlighted manual in the figure indicates the manual you are currently reading.

Figure 1 Gateway-ST Manuals



LKG−5089−93R

Figure 2 Gateway-CT Manuals



Access Routine Manuals

In addition to the gateway base communication system, Digital Equipment Corporation provides the following access routine products that use the DECnet /SNA gateways to access IBM systems. Each of these access routines has its own manual or set of manuals.

- **Bulk Data Transfer**
  - DECnet SNA Data Transfer Facility for OpenVMS
  - DECnet SNA Printer Emulator for OpenVMS
  - DECnet SNA Remote Job Entry for OpenVMS
- **Programming Interface**
  - DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS
  - DECnet SNA Application Programming Interface for OpenVMS
  - DECnet SNA 3270 Data Stream Programming Interface for OpenVMS
  - DECnet SNA ULTRIX 3270 Data Stream Programming Interface
- **Interactive Access**
  - DECnet SNA MS–DOS 3270 Terminal Emulator
  - DECnet SNA ULTRIX 3270 Terminal Emulator
  - DECnet SNA 3270 Terminal Emulator for OpenVMS
  - DECnet SNA VMS Distributed Host Command Facility
  - DECwindows DECnet SNA 3270 Terminal Emulator for ULTRIX
  - DECwindows DECnet SNA 3270 Terminal Emulator for OpenVMS
- **Office Information Transfer**
  - DECnet SNA DISOSS Document Exchange Facility
  - External Document Exchange with IBM DISOSS
  - VAX Message Router/P Gateway
  - VAX Message Router/S Gateway

OpenVMS Manuals

The following OpenVMS documents are also useful:

- *OpenVMS Networking Manual*
- *OpenVMS Network Control Program Manual*
- *OpenVMS System Messages and Recovery Procedures*
- *OpenVMS System Manager's Manual*

IBM Manuals

The following IBM documents are also useful:

- *SNA System Problem Determination Guide* (IBM Order No. G320-6016)
- *Advanced Communication Functions for VTAM, Operations* (IBM Order No. ST27-0612)
- *Advanced Communication Functions for VTAM, Diagnosis* (IBM Order No. ST27-0615)
- *ACF/VTAM Diagnosis Reference* (IBM Order No. SC27-0621)
- *ACF/SNA System Problem Determination Guide, Vol 1* (IBM Order No. GG24-1514)
- *ACF/SNA System Problem Determination Guide, Vol 2* (IBM Order No. GG24-1523)
- *Systems Network Architecture Formats* (IBM Order No. GA27-3136)
- *CICS/OS/VS Version 1 Release 7 Problem Determination Guide* (IBM Order No. SC33-0242)
- *CICS/OS/VS Version 1 Release 7 Data Areas* (IBM Order No. LY33-6035)
- *CICS/VS Version 1 Release 7 Diagnosis Reference* (IBM Order No. LC33-0243)
- *NetView Operation* (IBM Order No. SC30-3364)
- *NetView Messages* (IBM Order No. SC30-3365)
- *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information* (IBM Order No. GA22-6974)

Conventions

This manual uses the following conventions throughout:

Convention	Meaning
`SNANCP> SET LINE` *`line-id`*	In command examples, black monospaced type indicates system output. Red monospaced type indicates user input. In both command and syntax examples, uppercase letters represent text that you must enter exactly as shown. Lowercase letters in italics represent variables for which you must substitute specific information.
[*opt-arg*]	Square brackets enclose optional parts of a command.
{ ACTIVE LINES KNOWN LINES LINE *line-id* }	Braces indicate that you must choose only one of the command words or arguments in the list.
Return	A symbol with the word Return indicates that you press the Return key on the terminal keyboard. Unless otherwise stated, end every command line by pressing Return.
Ctrl/*x*	This symbol indicates that you press and hold down the key labeled Ctrl while simultaneously pressing another key (for example, Ctrl/C or Ctrl/T).

When you issue NCP and SNANCP commands, many component names, parameters, and qualifiers require additional information. In most cases, the syntax of this additional information follows a standard set of rules. (Exceptions to these rules are noted where appropriate.) All numeric values are in decimal form and range from 0 to 65,535 unless otherwise specified.

SNANCP and NCP commands use the following conventions:

Convention	Meaning
circuit-id	A string of characters whose exact syntax is that for a DECnet or SNA circuit identification. On DECnet nodes, circuit identification takes the following form: *dev-c* Where *dev* A device name. *c* A decimal number (0 or a positive integer) designating the device's hardware controller. On the Gateway node, the Ethernet circuit is identified by the string ETHERNET. SDLC circuits are identified by a string in the form SDLC-*n*, where *n* indicates the number of the circuit. Channel-attached circuits are identified by a string in the form CHAN-*n*, where *n* indicates the number of the circuit.
E-address	A string of 12 hexadecimal digits, represented by 6 bytes separated by hyphens (for example, 08-00-2B-08-E3-BE). The string indicates the Ethernet hardware address.
line-id	A string of characters whose exact syntax is that for a DECnet or SNA line identification. On DECnet nodes, line identification takes one of the following formats: *dev-c* *dev* A device name. *c* A decimal number (0 or a positive integer) designating the device's hardware controller. On the Gateway node, the Ethernet line is identified by the string ETHERNET. SDLC lines are identified as a string in the form SYN-*n*, where *n* is the number of the line. Channel-attached lines are identified as a string in the form CQ-*n*, where *n* indicates the number of the line.

Convention	Meaning
node-address	A numeric value ranging from 1.1 to 63.1023, composed of an area number to the left of the period followed by a node number to the right of the period. If the area number is not supplied, the area number of the executor node is used. The default area number for the executor is 1.
node-id	Either a *node-name* or a *node-address*.
node-name	A string of up to 6 alphanumeric characters containing at least 1 alphabetic character.

Abbreviations and Acronyms

This manual uses the following abbreviations and acronyms:

ACF/NCP	Advanced Communications Function with Network Control Program (also referred to as IBM NCP)
API	DECnet SNA Application Programming Interface for OpenVMS
APPC/LU6.2	DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS
CICS/VS	Customer Information Control System/Virtual Storage (also referred to as CICS)
DECnet NCP	Digital's Network Control Program (also referred to as NCP)
DDXF DECnet /SNA VMS DISOSS Document Exchange Facility	
DHCF DECnet /SNA VMS Distributed Host Command Facility	
DSPRINT	MVS/TSO/VTAM Data Set Print Facility
DTF	DECnet SNA Data Transfer Facility
EDE	External Document Exchange with DISOSS
GAS	Gateway Access Server
IBM NCP	IBM's Network Control Program
IMS/VS	Information Management System/Virtual Storage (also referred to as IMS)

JES2	Job Entry Subsystem 2
JES3	Job Entry Subsystem 3
LU	Logical unit
LU1	Logical unit type 1
LU2	Logical unit type 2
LU3	Logical unit type 3
LU6.2	Logical unit type 6.2
MVS	IBM’s Multiple Virtual Storage operating system
MR/S	VAX Message Router/S Gateway
MR/P	VAX Message Router/P Gateway
NCL	Network Control Language
NCP	Network Control Program. This manual uses NCP to refer to the Digital product and IBM NCP to refer to the IBM product.
PLU	Primary logical unit
PrE	DECnet SNA Printer Emulator for OpenVMS
PU	Physical unit
PU2	Physical unit type 2
RH	Request/response header
RJE	DECnet SNA Remote Job Entry for OpenVMS
RU	Request/response unit
SDLC	Synchronous Data Link Control
SLU	Secondary logical unit
SNA	IBM’s Systems Network Architecture
SNANCP	DECnet/SNA Gateway Network Control Program
SNAP	DECnet/SNA Gateway Management display utility
SNATRACE	DECnet/SNA Gateway Management protocol trace utility
SSCP	System services control point
3270 DS	DECnet/SNA 3270 Data Stream Programming Interface
3270 TE	DECnet/SNA 3270 Terminal Emulator
TH	Transmission header
TSO	IBM’s Time Sharing Option
VM	IBM’s Virtual Machine operating system
VSE/SP	Virtual Storage Extended/System Package

VTAM	Virtual Telecommunications Access Method

1 Overview

The DECnet SNA Gateway-ST (Synchronous Transport) and the DECnet SNA Gateway-CT (Channel Transport) are systems developed by Digital Equipment Corporation that allow bidirectional communication between an IBM® SNA network and a Digital DECnet™ network. The DECnet SNA Gateway-ST provides an interface to the IBM SNA network over SDLC lines and circuits. The DECnet SNA Gateway-CT is channel-attached directly to an IBM mainframe.

The Gateway systems enable nodes in a DECnet network that have the appropriate access routine software installed, to communicate with one or more hosts in an IBM SNA network. The Gateway systems translate the different communications protocols that exist between DECnet and SNA.

This book uses the term Gateway to refer to both the DECnet SNA Gateway-ST and the DECnet SNA Gateway-CT. Gateway-ST and Gateway-CT refer to the transport-specific product only.

The Gateway-ST hardware is the DEC MicroServer. This communications processor can be either a four-port or single-port model.

Note

Throughout this book, "MicroServer" refers to both the four-port DEC MicroServer (also known as the DEMSA) and the single-port DEC MicroServer-SP (also known as the DEMSB) unless otherwise stated.

The Gateway-CT hardware is the ChannelServer, a high-speed channel-to-Ethernet interface that is directly attached to an IBM channel. There are two processor models for this product: the DEC ChannelServer (also known as the DESNA) and the DEC ChannelServer II (also known as the DESNB).

Note

Throughout this book, "ChannelServer" refers to both the DESNA and DESNB models unless otherwise stated.

Both the Gateway-ST and the Gateway-CT utilize the following software:

- **Gateway software**: The communications software that controls the transfer and flow of information between the SNA network and the DECnet network. The Gateway software is installed on an OpenVMS-based system called the load host. The software is then down-line loaded into the Gateway.

  The Gateway software also includes management software. You must install the management functionality on the load host, and you may install it on any OpenVMS VAX, or OpenVMS AXP node that has DECnet connectivity to the Gateway. Digital Equipment Corporation strongly recommends that you install Gateway management software on any nodes that use the following program development access routines:

  – DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS

  – DECnet SNA 3270 Data Stream Programming Interface for OpenVMS

  – DECnet SNA Application Programming Interface for OpenVMS

- **Access routines**: The individual software products that use the Gateway to access programs and information on an IBM system. You can use various access routines depending on your application. For example, the DECnet SNA Data Transfer Facility for OpenVMS (DTF) lets you transfer files bidirectionally between OpenVMS systems and IBM systems.

1.1 DECnet SNA Network

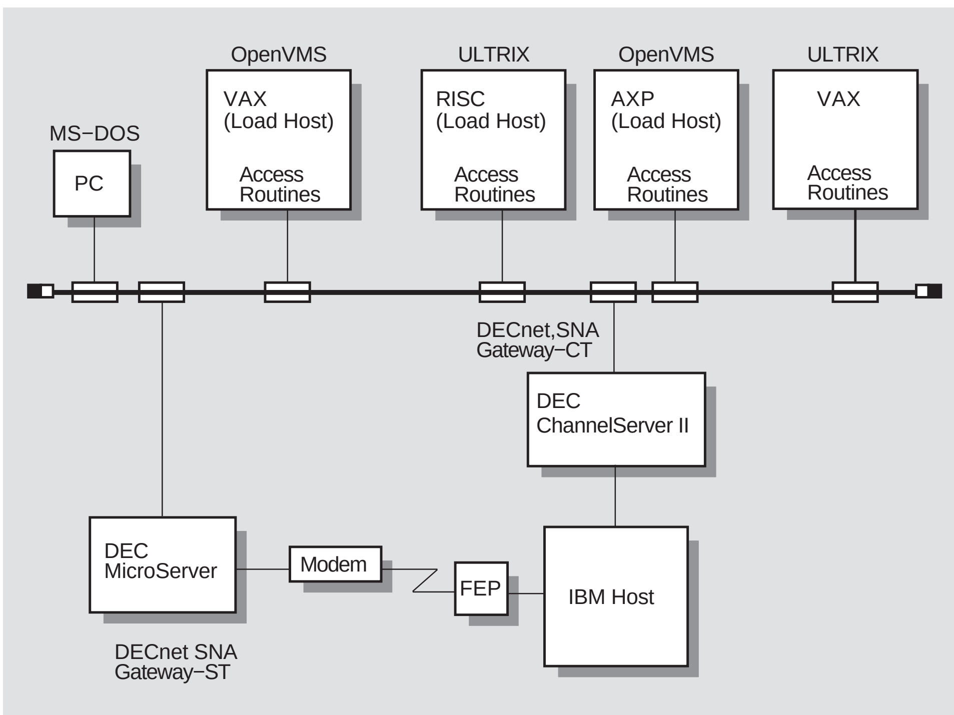
A DECnet SNA network is a data communications network that consists of the following:

- A DECnet network
- An IBM SNA network
- One or more Gateway systems

Data communications networks generally consist of components called nodes. The nodes are connected by communication links. The following sections describe the components of a DECnet SNA network.

Figure 1–1 shows a DECnet SNA network.

Figure 1–1 DECnet SNA Network



LKG−5015−93R

1.1.1 DECnet Network

A DECnet network with a Gateway includes the following types of nodes connected by communication links:

- **Gateway node**: A DEC MicroServer or a DEC ChannelServer into which the Gateway software is down-line loaded. The Gateway node performs the following functions:
  - Translates DNA protocol into SNA protocol
  - Provides SNA line connections. The MicroServer supports 4 lines; the MicroServer-SP supports one line; the ChannelServer supports one line.

- Provides servers for DECnet SNA OpenVMS products such as DHCF, RJE, 3270 TE, DTF, and customer-written programs that use the DECnet SNA programming interfaces

- **Gateway load host node**: A DECnet node that has the complete Gateway software kit, including managment software, installed. This node can perform the following functions:
  - Down-line load the Gateway software.
  - Bootstrap the Gateway node with the TRIGGER or LOAD commands.

Note

NCP LOAD is not supported when using the Gateway-CT; if you use the LOAD NODE command it will act as a TRIGGER NODE command.

  - Receive up-line dumps from the Gateway node.
  - Use Gateway management utilities such as SNANCP, SNAP, NETTRACE, and SNATRACE.
  - Configure the Gateway-ST or the Gateway-CT.
  - Log events from the Gateway node.
  - Run loopback tests on SDLC lines.

  This node can optionally run DECnet SNA access routine software, such as the DECnet SNA Data Transfer Facility (DTF) for OpenVSM.

- **Gateway access node**: Any DECnet node from which you can access the Gateway. The node must have one or more of the DECnet SNA access routines installed.

  A Gateway access node uses the Gateway to access information and programs on the IBM host.

You may install the Gateway management software on an access node. Digital recommends that you install the Gateway management software on all access nodes that use the DECnet SNA programming interface products.

With Gateway management software, an access node can perform the following functions:

- Use Gateway management utilities such as SNANCP, SNAP, NETTRACE, and SNATRACE.
- Receive event logging data from the Gateway node.

1.1.2 SNA Network

An SNA network consists of the following types of nodes interconnected by communications links:

- **IBM host node**: A computer system that is running the MVS, VM, or VSE/SP operating system and that provides centralized control and data processing functions for the SNA network. An IBM System/370, 303*x*, 308*x*, 309*x*, 43*xx*, 937*x* or later generation computer can be an IBM host node. Several host nodes can exist in an SNA network. Software on the IBM host typically includes the following:
  - Application programs that provide specific facilities. DISOSS™ is an example of an application program.
  - Application subsystems that provide functions and services to IBM application programs. CICS™ is an example of an application subsystem.
  - Job entry subsystems that provide functions that enable users at remote workstations to use the batch processing facilities of the IBM host. JES2 is an example of a job entry subsystem.
  - Telecommunications access methods that provide functions needed to transfer data between application subsystems in the host and remote systems. VTAM is an example of a telecommunications access method.
- **Communications controller**: A processor that carries out some of the control functions for the host node, for example, an IBM 3725. The main software in a communications controller is the Network Control Program with Advanced Communications Function (ACF/NCP). IBM's NCP is directed by VTAM or another access method in the IBM host. SDLC lines are typically connected through the communications controller.

Note

Do not confuse IBM's NCP with Digital's Network Control Program (NCP). Digital's NCP is a network management utility. This manual uses NCP to refer to the Digital product and uses IBM NCP to refer to the IBM product.

- **Cluster controller**: A processor that controls attached I/O devices; for example, a 3274 control unit that is part of the IBM 3270 Information Display System (IDS). A cluster controller is a physical unit (PU) type 2.

For more information on SNA, see the appropriate IBM documentation.

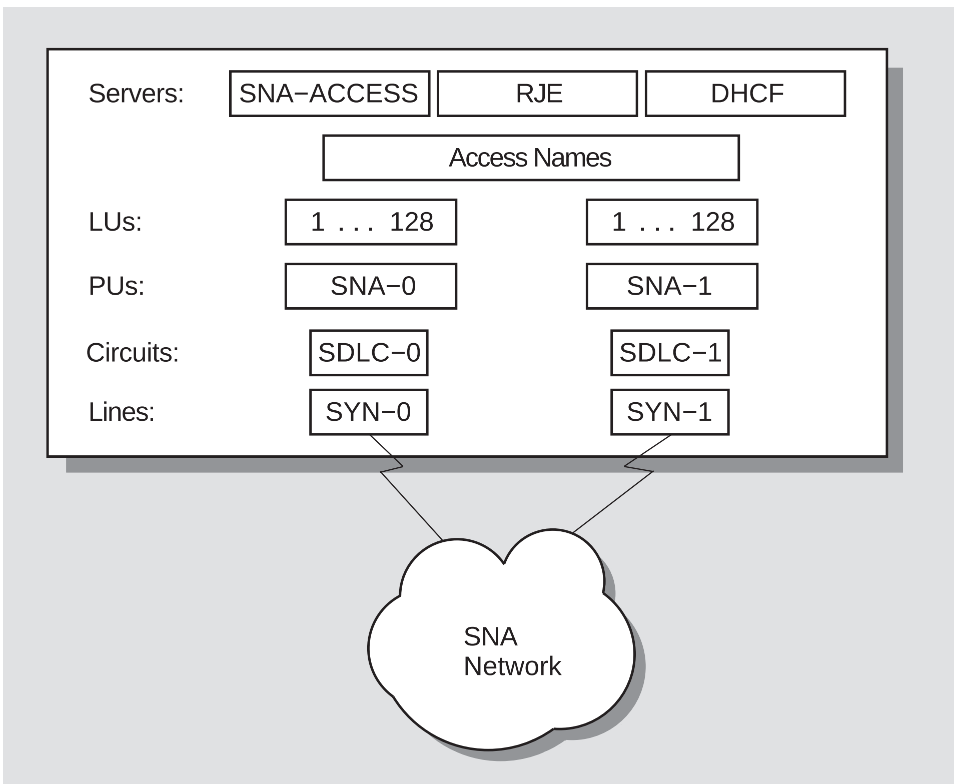
1.1.3 Gateway Node

The Gateway node connects DECnet and SNA networks. The Gateway node is a communications system based on the Gateway software and either the DEC MicroServer or DEC ChannelServer. To the DECnet network, the Gateway node is a DECnet/VAX end node. To the SNA network, the Gateway node is a cluster controller node that is a physical unit (PU) type 2. Section 1.1.3.1 describes the Gateway node components when using the Gateway-ST. Section 1.1.3.2 describes the Gateway node components when using the Gateway-CT.

1.1.3.1 Gateway-ST Components

Figure 1–2 shows a Gateway node with two SDLC lines attached to the SNA network. The components are described following the figure.

Figure 1–2 DECnet SNA Gateway-ST Node



LKG−4153−93R

SDLC lines are the physical links the Gateway-ST uses to transfer data to the IBM communications controller. Depending on the speed of the SDLC lines, the Gateway-ST can provide up to four links to IBM systems (up to four on the MicroServer; one on the MicroServer-SP).

SDLC circuits control the SDLC protocol on the line. You must define a line before you define a circuit to attach to that line. Each line supports a single circuit.

PUs (physical units) provide the low-level communications services for the node. Each circuit supports a single PU. You must define a circuit before you can define a PU for that circuit.

LUs (logical units) allow end users to access other LUs in an SNA network. A certain number of LUs are defined in an SNA host for use by different types of applications. When you are defining a PU, you define LUs for the Gateway as subaddresses of the PU.

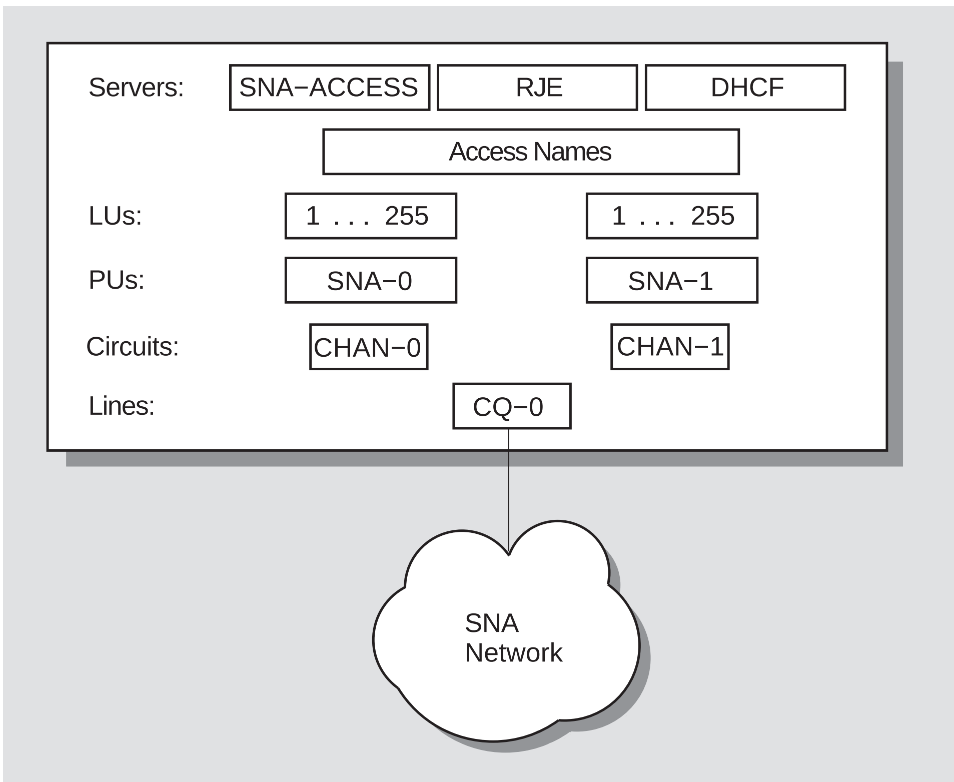
Access names represent a list of parameters that can be used to establish a session with an application running on the IBM system. You can define access names at any time.

Servers provide the services needed by the DECnet SNA access routines. For example, you need the SNA-ACCESS server if you want to use the DECnet SNA DDXF, DTF, MR/P, MR/S, 3270 TE, API, APPC/LU6.2, 3270 DS, or PrE access routines. Servers are always defined.

1.1.3.2 Gateway-CT Components

Figure 1–3 shows a Gateway node with a channel interface attached directly to an IBM host node. The Gateway components are described following the figure.

Figure 1–3 DECnet SNA Gateway-CT Node



LKG−4152−93R

The channel is the physical link the Gateway-CT uses to transfer data to the IBM system. Devices on the channel share the physical link. The channel can be a block multiplexer, byte multiplexer, or selector channel.

The channel is referred to as a channel-attached line in the Gateway-CT and Gateway management software. The Gateway-CT supports a single channel-attached line.

Channel-attached circuits control the 3274 datalink protocol on the channel. A channel-attached line supports several circuits. Each circuit corresponds to one address on the IBM channel. You must define a channel-attached line before you define a circuit to attach to that line.

PUs (physical units) provide the low-level communications services for the node. Each circuit supports a single PU. You must define a circuit before you can define a PU for that circuit.

LUs (logical units) allow end users to access other LUs in an SNA network. A certain number of LUs are defined in an SNA host for use by different types of applications. When you are defining a PU, you define LUs for the Gateway as subaddresses of the PU.

Access names represent a list of parameters that can be used to establish a session with an application running on the IBM system. You can define access names at any time.

Servers provide the services needed by the DECnet SNA access routines. For example, you need the SNA-ACCESS server if you want to use the DECnet SNA DDXF, DTF, MR/P, MR/S, 3270 TE, API, APPC/LU6.2, 3270 DS, or PrE access routines. Servers are always defined.

1.2 Managing a DECnet SNA Network

As a manager of a DECnet SNA network, you must understand three areas:

- The DECnet network
- The IBM SNA network
- The Gateway system

Usually, you will share the task of managing the components in a DECnet SNA network with other people. Normally one person acts as coordinator to ensure that all components are properly configured for Gateway operations.

1.2.1 DECnet Components

Day-to-day management of DECnet components involves setting line states, monitoring local and remote counters, and performing loopback tests. The standard management of DECnet components is documented in the *OpenVMS Networking Manual* and *OpenVMS Network Control Program Manual*.

This manual documents only those DECnet management tasks that are related to using the Gateway to access IBM system resources. This information includes the commands that you issue from a DECnet node to configure, monitor, and control Gateway components.

To eliminate confusion, Digital suggests designating one system manager in the network to manage the Gateway, because any DECnet node system manager can perform Gateway management tasks if the management software is installed on that node.

1.2.2 SNA Components

Management of IBM components (for example, nodes, front-end processors, and lines within the IBM network) is done with IBM-supplied utilities according to the policies of the IBM site.

Certain parameter values of IBM communications software must correspond to Gateway parameter values before the IBM system can exchange data with a DECnet network. Instruct the IBM systems programmers at your site to read the *DECnet SNA Gateway for Channel Transport Guide to IBM Parameters* or the *DECnet SNA Gateway for Synchronous Transport Guide to IBM Parameters* for information on how to set these parameters for IBM system generation.

1.2.3 Managing Gateway Components

When managing a Gateway system, you must understand the hierarchy of Gateway components so that you can define or change them in the proper order. The following components must be defined in the order listed:

1. Lines
2. Circuits
3. PUs and their corresponding LUs

Access names can be defined in any order. Servers are always defined.

Managing Gateway components, such as the lines and circuits from the Gateway to both the DECnet node and the IBM system, requires specialized management procedures and utilities. The following utilities can help you manage your system:

- **SNANCP**: The DECnet SNA Gateway Network Control Program is an interactive utility that allows you to manage the Gateway node and the components that communicate with the IBM system.
- **DECnet NCP and NCL**: You can only use NCP commands to manage the Gateway node and Gateway components that communicate with DECnet OpenVMS VAX and OpenVMS AXP nodes. You can use NCP or NCL commands to prepare the down-line load information for the Gateway and to run DECnet loopback tests.
- **SNAP**: SNAP is a display utility that provides dynamic information about the current communications resources of the Gateway.
- **NETTRACE and SNATRACE**: These DECnet SNA protocol trace utilities are VMS host-based utilities that allow you to collect information and analyze the flow of data between the Gateway and the IBM system.

1.2.3.1 Gateway Management Tasks

The following is a list of tasks you perform to manage a DECnet SNA network:

- Install the Gateway hardware
  - When using the Gateway-ST, install the DEC MicroServer (see *Installing the MicroServer* or *Installing the DEC MicroServer-SP*)
  - When using the Gateway-CT, install the DEC ChannelServer (see the *DEC ChannelServer Hardware Installation Guide* or *DEC ChannelServer II Installation*)
- Install and configure the Gateway software
  - When using the Gateway-ST, see *DECnet SNA Gateway for Synchronous Transport Installation*
  - When using the Gateway-CT, see *DECnet SNA Gateway for Channel Transport Installation*
- Down-line load the Gateway software from the DECnet load host to the Gateway node
  - When using the Gateway-ST, see *DECnet SNA Gateway for Synchronous Transport Installation*
  - When using the Gateway-CT, see *DECnet SNA Gateway for Channel Transport Installation*
- Secure the Gateway system (see Chapter 2)
- Manage lines (see Chapter 4)
- Manage circuits (see Chapter 5)
- Manage PUs (see Chapter 6)
- Manage LUs (see Chapter 7)
- Manage access names (see Chapter 8)
- Manage servers (see Chapter 9)
- Control which events are logged (see Chapter 10)
- Check the meaning of events that are logged (see Appendix D)

You need to consider many factors when managing the Gateway components. The following sections describe several important points about managing a DECnet SNA network.

1.2.3.2 Getting HELP

There are several ways to get on-line help for the Gateway.

For help on Gateway management at the DCL level, enter the following command:

```
$ HELP SNA_GM
```

To get help while using the NCP utility, enter HELP after the NCP prompt:

```
$ RUN SYS$SYSTEM:NCP
NCP>HELP
```

To get help while using the NCL utility, enter HELP after the NCL prompt:

```
$ RUN SYS$SYSTEM:NCL
NCL>HELP
```

To get help while using the SNANCP utility, enter HELP after the SNANCP prompt:

```
$ RUN SYS$SYSTEM:SNANCP
SNANCP> HELP
```

To get help while using the SNAP utility, press the H key.

To get help while using the NETTRACE utility, enter HELP after the NETTRACE prompt:

```
$ TRACE
NETTRACE> HELP
```

To get help while using the SNATRACE utility, enter HELP after the SNATRACE prompt:

```
$ RUN SYS$SYSTEM:SNATRACE
SNATRACE> HELP
```

1.2.3.3 Required Privileges

You need the following privileges to manage the Gateway:

- TMPMBX to create a temporary mailbox.
- NETMBX to create a network mailbox. This privilege is required for all DECnet users running network programs.
- SYSPRV to edit the configuration files.
- OPER to issue NCP service commands.

Refer to your OpenVMS system manager's documentation for more information about these privileges.

1.2.3.4 Configuring the Gateway

There are two kinds of configuration: initial configuration and reconfiguration. The following sections describe how to perform each type of configuration.

1.2.3.4.1 Initial Configuration For the initial configuration of a Gateway system, you run a configuration command procedure after installing Gateway software. This procedure creates other command procedures that will configure your Gateway each time it is reloaded. See *DECnet SNA Gateway for Channel Transport Installation* or *DECnet SNA Gateway for Synchronous Transport Installation* for a complete description of the Gateway configuration procedure. The initial configuration is simply a way to get your Gateway up and running. Modify the configuration files to meet your exact system requirements.

When you run the configuration procedure, the procedure prompts you for a small number of site-specific values. In addition to default information, the configuration procedure uses the information that you provide to initially define components for your Gateway system. The definitions are executed at load time.

1.2.3.4.2 Reconfiguration After the Gateway is initially configured and loaded, you can customize the configuration for your site. The Gateway software provides SNANCP and NCP commands that you can enter to interactively change the characteristics of your system. Subsequent chapters in this book discuss how to use these commands to tailor your system.

When you enter commands interactively, the definitions or changes you make take effect immediately. However, these definitions or changes stay in effect only while the Gateway is running. When the Gateway is reloaded, the values in the command procedures that you created during the initial configuration are used.

Digital suggests the following procedure for reconfiguring your Gateway system. Use interactive commands to make changes to the Gateway system while it is running. If you want to keep those changes the next time the Gateway is loaded, edit the appropriate command procedure to change the previously defined values:

- To change the definitions of the SNA components in the Gateway database, edit the SNANCP commands in the following file:

  SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_SNA.COM

- To change the definitions of the DECnet components in the Gateway database, edit the NCP commands in the following file:

  SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_DNA.COM

The values in the preceding command procedures are used the next time the Gateway is reloaded.

Note

If you have any special site-specific configuration requirements, such as LU authorization information, that you need to include as part of your Gateway start up procedure, add it to one of the appropriate command procedures. Do not put any site-specific start up information in the command procedure called SNAINI.COM. Digital Equipment Corporation reserves the right to modify the configuration procedure, and may not support this command procedure in future releases of Gateway software.

1.2.3.5 Event Logging

You can specify that any node running Gateway management software is to receive event logging information generated by the Gateway. The event logger lets you do the following:

- Send event information to OPCOM for display on operator terminals and in the operator log.
- Store information in a file, or display information on a dedicated terminal, in text form.
- Store information in a file in binary form. This file can be used by Digital Customer Services to help you locate problems.

Event logging is described in *DECnet SNA Gateway for Channel Transport Installation* and *DECnet SNA Gateway for Synchronous Transport Installation*. The *OpenVMS Networking Manual* provides you with information about DECnet event logging. See Chapter 10 for information on how to use the SNANCP utility to control which Gateway event messages are recorded and where they are recorded. The Gateway event messages are listed in Appendix D.

1.2.3.6 Error Messages

Although you define components for the Gateway using SNANCP, any of the following utilities can report errors in the configuration:

- SNANCP (messages are listed in Appendix C).
- SNAP (messages are listed in Appendix C).
- NETTRACE (messages are listed in Appendix C).

- SNATRACE (messages are listed in Appendix C).
- DECnet (messages are listed in the *OpenVMS System Messages and Recovery Procedures*).
- SNAEVL (messages are listed in Appendix D).
- A DECnet SNA access routine, such as the 3270 TE (see the appropriate access routine documentation for an explanation of the message).
- The IBM system (see the appropriate IBM documentation).

1.2.3.7 Diagnostic Tools

You can use the following diagnostic tools to isolate problems in the Gateway hardware or software:

- **Loopback tests**, which isolate problems in the network by sending test messages over a particular data communications path and looping them back to the originating node. Refer to *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)* or *DECnet SNA Gateway for Synchronous Transport Problem Solving* for information on how to use loopback tests.
- **Traces**, which show the SNA protocol exchange between the Gateway node and an IBM node. NETTRACE and SNATRACE are two Gateway management utilities which work from the Gateway to isolate problems. You can also use IBM traces which work from the IBM system to isolate problems or to show the sequence of events in a session. Chapter 11 and Chapter 12 contain information on NETTRACE and SNATRACE, respectively. Refer to Appendix C for an explanation of NETTRACE and SNATRACE messages.
- **Up-line dumps**, which are files containing memory dump data created if the Gateway system crashes. The up-line dump database is created when you configure the Gateway. These files are used by Digital support personnel to solve problems with your system.
- **DEC MicroServer remote console commands**, which allow you to control the operation of the system, display information, and test the communications lines. See *DECnet SNA Gateway for Synchronous Transport Problem Solving* for information on using the MicroServer remote console commands.
- **DEC ChannelServer remote console commands**, which allow you to control the operation of system and display information. See *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)*) for information on using the ChannelServer remote console commands.

2

Gateway Security

Because a Gateway system is part of an open network, any node in the DECnet network can potentially access your Gateway. As part of the security for your Gateway system, all nodes on the same Ethernet as the Gateway should follow proper security measures. For information on OpenVMS security, see the *Guide to OpenVMS System Security*. For information on protecting DECnet components, see the *OpenVMS Networking Manual*.

The Gateway software enables you to protect your network and the Gateway system itself. This chapter provides an overview of the types of access control you can use to achieve the level of security your site needs. The Gateway node implements access control in much the same way as any other DECnet node. A verification program compares all incoming access control information against information provided by the Gateway configuration database. If the information matches, the verification program allows access to the Gateway. If the information does not match, access is denied.

You can control access to any of the following Gateway components:

- DEC MicroServer and remote console commands
- DEC ChannelServer and remote console commands
- DECnet and SNA components in the Gateway
- Gateway system files
- Individual LUs
- RJE files

2.1 Securing the Gateway Hardware

The Gateway system is not secure if unauthorized personnel have access to either the DEC MicroServer and its remote console commands or to the DEC ChannelServer and its remote console commands. For example, individuals can interfere with the Gateway system by unplugging the DEC ChannelServer or by pushing the DUMP switch on the DEC MicroServer control panel. To prevent unauthorized use of the DEC MicroServer or the DEC ChannelServer, Digital Equipment Corporation recommends that you place them in a limited-access area.

2.2 Securing the Remote Console

The DEC MicroServer and the DEC ChannelServer provide remote console commands for controlling and monitoring the system. (For information on using the remote console commands, see *DECnet SNA Gateway for Synchronous Transport Problem Solving* or *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)*. Because some of the remote console commands can be used to compromise the security of the Gateway, restrict access to the remote console commands by defining a service password.

Access to the ChannelServer remote console may be restricted by defining a password with remote console commands. These commands are implemented by software in the Gateway-CT. You can use them only if the Gateway-CT is running.

2.2.1 Defining a Service Password

A service password is a hexadecimal number that users must specify to do the following (for DECnet/VAX):

- Use the NCP CONNECT command to attach a terminal as a remote console
- Use the NCP TRIGGER or LOAD commands to load the Gateway software from a load host

Note

The NCP LOAD command can only be used with the Gateway-ST.

And for DECnet/OSI:

- Use the NCL SET HOST/MOP command to attach a terminal as a remote console

- Use the NCL BOOT MOP CLIENT command to load the Gateway software from a load host

When you first install a DEC MicroServer or a DEC ChannelServer, no password has been previously defined. This allows you to perform the initial startup of the hardware and load your software system without having to know a preset password.

2.2.1.1 MicroServer

Once your MicroServer is running, you can define a service password by attaching a terminal as a remote console with the following command (for DECnet/VAX):

```
NCPCONNECT NODE gateway-node-name
```

and for DECnet/OSI:

```
$ SET HOST/MOP gateway-node-name VERIFICATION=%xverification
```

Replace *verification* with up to 16 hexadecimal digits; for example:

```
$ SET HOST/MOP gateway-node-name VERIFICATION=%x1212121212121212
```

When you see the prompt for the remote console (>>>), use the SET PASSWORD command to define a service password. The command syntax is as follows (for both DECnet/VAX and DECnet/OSI):

```
>>>SET PASSWORD password
```

Replace *password* with up to 16 hexadecimal digits; for example:

```
>>>SET PASSWORD FEFEFEFEFEFEFEFE
```

Digital recommends that your service password be at least six hexadecimal digits in length.

If you do not enter a password after the command but merely press Return, the system prompts you for a password. The system does not echo the password you enter. The system then prompts you for verification. Enter the password again.

As with all passwords, it is wise to change the service password regularly to maintain security. To change the password, simply use the SET PASSWORD command.

To show the password, use the following command:

```
>>>SHOW PASSWORD
```

When you have finished using the remote console, press Ctrl/D for DECnet Phase IV, and Ctrl / for DECnet/OSI. This returns you to the host, and the NCP prompt (Phase IV) or $ prompt (DECnet/OSI) appears on your terminal.

2.2.1.2 ChannelServer

When using the ChannelServer, a service password is established when running SNACSA$CONFIGURE. In Part 3 of the Gateway-CT configuration, you define a service password by answering the following prompt:

```
DECnet service password (up to 16 hex digits) [none]:
```

The password is included in the file SNAGATEWAY_*gateway-node-name*_DNA.COM. The password takes effect when the Gateway-CT is loaded. You can change the password by editing the following line within the command file:

```
$ CONSOLE "SET PASSWORD 2267118"
```

You also have the option of defining a service password using the remote console facility. However, the password remains in effect only until the Gateway-CT is rebooted.

To permanently change the password, you must edit the file, SNAGATEWAY_*gateway-node-name*_DNA.COM.

You can define a service password by attaching a terminal as a remote console with the following command (for DECnet/VAX):

```
NCP>CONNECT NODE gateway-node-name
```

and for DECnet/OSI:

```
$ SET HOST/MOP gateway-node-name VERIFICATION=%xverification
```

Replace *verification* with up to 16 hexadecimal digits; for example:

```
$ SET HOST/MOP gateway-node-name VERIFICATION=%x1212121212121212
```

After you have connected your terminal as a remote console, press the Return key until the remote console prompt appears.

When you see the prompt for the remote console (*node-name*>), use the SET PASSWORD command to define a service password. The command syntax is as follows:

```
node-name>SET PASSWORD password
```

Replace *password* with up to 16 hexadecimal digits; for example:

```
node-name>SET PASSWORD FEFEFEFEFEFEFEFE
```

Digital recommends that your service password be at least six hexadecimal digits in length. As with all passwords, it is wise to change the service password regularly to maintain security.

To show the password, use the following command:

```
node-name>SHOW PASSWORD
```

When you have finished using the remote console, press Ctrl/D for DECnet Phase IV, and Ctrl / for DECnet/OSI. This returns you to the host, and the NCP prompt (Phase IV) or $ prompt (DECnet/OSI) appears on your terminal.

2.2.2 Clearing the Service Password

There are circumstances in which you may want to change the password on either the MicroServer or ChannelServer. Sections 2.2.2.1 and 2.2.2.2 describe how to clear the service password on each system.

2.2.2.1 DEC MicroServer

If you forget the password when using the DEC MicroServer, you can clear the password by unplugging the DEC MicroServer and then powering it up in the following way:

1. Hold in the DUMP switch.
2. Insert the power cord.
3. Release the DUMP switch when the display shows a value of 1.

Once you have performed these three steps, you can set a new password using the SET PASSWORD remote console command.

2.2.2.2 DEC ChannelServer

You can clear the service password for the ChannelServer by attaching a terminal as a remote console with the following command for DECnet/VAX):

```
NCP>CONNECT NODE gateway-node-name
```

and for DECnet/OSI:

```
$ SET HOST/MOP gateway-node-name VERIFICATION=%x
```

When you see the prompt for the remote console (*node-name*>), use the CLEAR PASSWORD command to clear the service password. The command syntax is as follows:

```
node-name>CLEAR PASSWORD
```

To permanently clear the password, edit the file SNAGATEWAY_*gateway-node-name*_DNA.COM, and change the password to zero. (Zero means no password is defined.) For example:

```
$ CONSOLE "SET PASSWORD 0"
```

2.3 Securing DECnet and SNA Components

Like many OpenVMS products, the Gateway uses both privileged and nonprivileged usernames and passwords to control access to certain functions. Unlike many OpenVMS products (which use privileged and nonprivileged names and passwords as defaults in the absence of explicit access control information), the Gateways require that users specify privileged or nonprivileged access control information, if such information is defined.

2.3.1 Privileged Username and Password

Gateway systems use a privileged username and password to control access to the following functions:

- NCP CLEAR, LOOP, SET, and ZERO commands
- SNANCP CLEAR, LOOP, SET, SHOW LU (for authorization information), and ZERO commands
- NETTRACE commands
- SNATRACE commands

When you configure the Gateway, you can set up a privileged username and password for the system. This information is kept in the file:

SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_DNA.COM

You must ensure that the protection on this configuration file and its directory is set so that unauthorized users cannot read the information in the file. The file directory and protection is set properly when you run the configuration procedure. If you edit this configuration file, verify that any new file has the proper protection also. See Table 2–1 for the correct protection for the file.

If you do not set a privileged username and password during configuration, you can set them later with NCP commands. First, specify the Gateway (as the executor node) and then specify the privileged information for the system.

```
NCP>SET EXECUTOR NODE gateway-node-name
NCP>SET EXECUTOR PRIVILEGED USER name PASSWORD password
```

You can change the privileged username and password by using the SET EXECUTOR PRIVILEGED USER command again.

Once you have made an interactive change to the privileged information with NCP commands, edit the configuration file, SNAGATEWAY_*gateway-node-name*_DNA.COM. The information in this configuration file is used as the privileged information the next time the Gateway is rebooted.

2.3.2 Nonprivileged Username and Password

You can control access to Gateway functions other than those mentioned in Section 2.3.1 (for example, the SNAP utility, access to the Gateway by DECnet SNA OpenVMS access routines, and SNANCP and NCP SHOW commands) by requiring a nonprivileged username and password.

You cannot define a nonprivileged username and password unless you have already defined a privileged username and password.

To define a nonprivileged username and password, use NCP commands. First, specify either Gateway as the executor node; then specify the nonprivileged information for the system. For example:

```
NCP>SET EXECUTOR NODE ALACK
NCP>SET EXECUTOR NONPRIVILEGED USER FRED PASSWORD BARNEY
```

Once you have defined a nonprivileged username and password, these must then be supplied when using DECnet SNA access routines. For example, issue the following command to use the DECnet SNA 3270 Terminal Emulator for OpenVMS:

```
$ SET HOST/SNA/ACCESS=TSO ALACK"FRED BARNEY"
```

Note

Not all of the DECnet SNA OpenVMS access routines support nonprivileged access control. If you define nonprivileged access control for your system, access routines that do not support nonprivileged access control will not be able to use the Gateway. Refer to the documentation for the individual products to see which ones support nonprivileged usernames and passwords.

2.4 Securing System Files

The system image files, configuration files, and dump files contain important system information. To have a secure system, you must ensure these files are given the proper protection when the system creates these files. If you alter these files or change their default location, you must ensure that the files are properly protected. Table 2–1 shows the correct protection for the directory SYS$COMMON:[SNA$CSV] and system files:

Table 2–1 Protection of Gateway System Files

File	Protection
SNA$CSV.DIR	S:RWE, O:RWE, G, W
SNACST*nnn*.SYS (Applies to the DEC MicroServer)	S:RWED, O:RWED, G, W
SNACSA*nnn*.SYS (Applies to the DEC ChannelServer)	S:RWED, O:RWED, G, W
SNAINI.COM	S:RWED, O:RWED, G, W
SNAGATEWAY_*gateway-node-name*_SNA.COM	S:RWED, O:RWED, G, W
SNAGATEWAY_*gateway-node-name*_DNA.COM	S:RWED, O:RWED, G, W
DEFINE_*gateway-node-name*_ON_*load-host-name*.COM	S:RWED, O:RWED, G, W

Dump files must also be protected. If you create dump files in SYS$COMMON:[SNA$CSV], the system will give the files the proper ownership and protection.

The ownership of system files is an important part of securing a Gateway system. When you install the Gateway software, the installation procedure creates a directory (SYS$COMMON:[SNA$CSV]) and a default account (SNA$CSV) for the files. [SNA$CSV] must own the files in the directory SYS$COMMON:[SNA$CSV].

During installation, you are asked to specify a numeric user identification code (UIC) for the account. A default UIC of [56,56] is supplied. If you choose to use a different UIC, make sure that you specify a UIC that is unique, that is not a system UIC, and that is not in a group with any other UICs.

When the SNA$CSV account is created, it is given a randomly generated password. You can use AUTHORIZE to change this password at any time.

During the configuration of the Gateway, the system creates a proxy account entry on the load host associated with the user SNA$GO on the Gateway node (*gateway-node-name*::SNA$GO). The proxy for SNA$GO is SNA$CSV. The remote user SNA$GO receives the same access rights as user SNA$CSV on the load host. This proxy account is used at initialization time to read the configuration files that reside on the load host.

2.5 Securing LUs

In addition to the types of access control already described, you can create authorization entries that limit access to Gateway functions on a logical unit (LU) basis. You can specify that only certain nodes, users, and/or terminals can have access to specific LUs.

The following command specifies that only the SYSTEM user on node THYME can access LU SNA-0.1 (the authorization identification is SYS_AUTH:

```
SNANCP> SET LU SNA-0.1 AUTHORIZATION SYS_AUTH NODE THYME USER SYSTEM
```

By default, there is no authorization information. When there is no authorization information for an LU, the LU is not secure and anyone is allowed to use it.

For more information on LU authorization, see Chapter 7.

2.6 Securing RJE Files

The RJE server in the Gateway must read RJE input files. Since the RJE server is viewed as a remote user when it tries to access files, you must protect the RJE files in a way that allows the RJE server to access them.

Depending on the type of security you want for your Gateway system, you can choose one of the following methods:

- Make the RJE input files world readable.

  This saves the time and effort required to enter access control information in the DCL SUBMIT/SNA command. However, all users on your system (and possibly other nodes in the network) can read and copy your RJE files.

- Require access control information for the files.

  This protects your files from users who do not have the access control information. You must use care though, because usernames and passwords are passed around the network and are often stored in command procedures. Also, all users, even the owner of the files, must include access control information in the SUBMIT/SNA command.

- Create a proxy account for the Gateway node.

The proxy login capability increases security by minimizing the need to specify explicit access control information in node specifications passed over the network or stored in command procedures. However, you must create a proxy account that allows the Gateway node to assume the same file access rights as the proxy account.

To create a proxy account, use AUTHORIZE to create or modify the network user authorization file, NETPROXY.DAT. For the Gateway, you must create a proxy account entry for the user on the Gateway node. You can set up a proxy account specifically for RJE file access, or you can use an existing account.

Invoke AUTHORIZE and use the ADD command to set up a proxy account on each node that accesses the Gateway:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD/PROXY gateway-name::DAP proxy-account-name/DEFAULT
```

In the following example a proxy account SEALS is created:

```
UAF> ADD/PROXY ALACK::DAP SEALS/DEFAULT
```

Once you have created the proxy account, the RJE server assumes all the access rights of the local user SEALS. You must ensure that user SEALS can read all the RJE input files. You can grant this access to SEALS in any of the following ways:

- You can put SEALS in the same UIC group as all other RJE users and then set the protection on the files to GROUP:R. All other members of the group can then read your files.
- You can assign SEALS a privilege such as READALL so that SEALS can access all the files regardless of the file protection.
- You can use the Access Control List (ACL) Editor to create an access control list for the RJE input files that explicitly grants read access to SEALS.

Although the Gateway manager sets up a proxy account, the user controls (by the command he or she enters) which method of access control is used.

If the user provides explicit access control information, the system uses that information. The following is an example of a command with access control information:

```
$ SUBMIT/SNA THYME"username password"::filename.JCL
```

If the user specifies explicit null access control, the system uses the default FAL (file access listener) access. The following is an example of a command that specifies null access control:

```
$ SUBMIT/SNA 0""::filename.JCL
```

If the user does not provide access control information, the system checks for and attempts to use a proxy account if one is present. If a proxy account exists, the user SEALS in the preceding example must have read access to the files. If there is no proxy account, the system uses the default FAL access. This means that the world must have read access to the files. The following is an example of a command that uses the proxy account:

```
$ SUBMIT/SNA filename.JCL
```

3

Gateway Management Utilities

Gateway software provides the following utilities for managing and monitoring Gateway operations:

- **SNANCP**—SNANCP is an interactive command utility that allows you to manage the Gateway components that communicate with the IBM system.
- **Selected NCP commands**—The Gateway supports a subset of DECnet NCP commands to allow you to manage DECnet components that communicate with the Gateway.
- **SNAP**—SNAP is a display utility that provides information on the current communications resources of the Gateway.
- **NETTRACE**—NETTRACE is a command utility that allows you to collect and analyze information on the flow of data between the Gateway and the IBM system. (NETTRACE is described in Chapter 11.)
- **SNATRACE**—SNATRACE is a command utility that allows you to collect and analyze information on the flow of data between the Gateway and the IBM system. (SNATRACE is described in Chapter 12.)

This chapter introduces the SNANCP, NCP, and SNAP utilities. It describes the functions of each utility and explains how to start the utilities, enter commands, and exit from the utilities.

Individual SNANCP and NCP commands are presented in Chapters 4 through 10 to demonstrate managing Gateway components such as lines, circuits, PUs and LUs.

Chapter 13 and Chapter 14 fully describe the SNANCP and NCP commands that are used with the Gateways.

3.1 SNANCP

The DECnet SNA Gateway Network Control Program (SNANCP) enables you to manage the Gateway components that communicate with the IBM system. You use SNANCP commands to define, monitor, and control the components and their parameters.

The values you define during the installation and configuration of the Gateway are the initial values for the parameters. Once the Gateway is running, you can redefine and manage components with SNANCP commands. The values you specify using SNANCP commands stay in effect until one of the following occurs:

- You change the values with other interactive SNANCP commands.
- You edit the SNANCP commands in the SNA configuration file and reboot the Gateway. The default values for the Gateway SNA components are stored in the following file:

  SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_SNA.COM

  The values in this file are used when the Gateway is rebooted.

3.1.1 Starting SNANCP

To start SNANCP, you can enter the following DCL command:

```
$ RUN SYS$SYSTEM:SNANCP
```

The following prompt will appear on your screen:

```
SNANCP>
```

Another way to invoke SNANCP is by defining it as a foreign command. You can add the following definition to your LOGIN.COM command procedure:

```
$ SNANCP == "$SNANCP"
```

Then, you can invoke SNANCP without using the RUN command.

3.1.2 Specifying the Gateway Node

Before you can use any SNANCP commands to manage the Gateway, you must specify the name of the Gateway node you will be using. Specify the name with the USE SYSTEM command. You can enter privileged or nonprivileged usernames and passwords with the USE SYSTEM command. (For information on privileged and nonprivileged information, see Chapter 2.) There are two ways to enter usernames and passwords:

```
SNANCP> USE SYSTEM gateway-name [USER username PASSWORD password]
```

or

```
SNANCP> USE SYSTEM gateway-node-name["username password"::]
```

You do not need to specify the USE command if you have defined the logical name SNAGM$GATEWAY to provide a default Gateway node specification to SNANCP.

You can define the privileged username and password for the Gateway when you run the configuration procedure that is part of the installation procedure (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). If a privileged username and password are defined, you must specify this information with the USE SYSTEM command if you want to use the SNANCP CLEAR, LOOP, SET, SHOW LU (for authorization information), and ZERO commands. If you do not specify a privileged username and password, you can use only the SHOW, EXIT, SPAWN, and HELP commands.

If you want to use a different Gateway node as the executor for SNANCP commands, enter the USE SYSTEM command again and specify a new Gateway node:

```
SNANCP> USE SYSTEM THYME USER BRETT PASSWORD SECRET
```

3.1.3 SNANCP Command Syntax

An SNANCP command can have the following parts: a command verb, a component name (and possibly a component qualifier), one or more parameters, and one or more qualifiers, for example:

Command	Component Name	Parameter	Qualifier
SHOW	LINE *line-id*	CHARACTERISTICS	TO *file-spec*
	KNOWN LINES	COUNTERS	
	ACTIVE LINES	STATUS	
		SUMMARY	

For each command, you must supply a command verb and a component name. Optionally, you can supply one or more parameters from the parameter list and one or more qualifiers from the qualifier list.

3.1.4 Entering SNANCP Commands

Separate SNANCP verbs and parameters by spaces or tabs. To continue a long command to the next line, use a hyphen as the line continuation character. For example:

```
SNANCP> SET LINE SYN-0 STATE ON DUPLEX HALF SIGNALLING -
_SNANCP> NORMAL MODEM TYPE NORMAL
```

Use an exclamation point (!) to designate a comment line.

SNANCP also provides command line editing and prompting mode to help you enter commands.

3.1.4.1 Command Line Editing

SNANCP supports command line editing and recall. SNANCP saves and allows you to access up to 20 of your most recently entered commands. (It saves fewer than 20 commands if your commands are very long.)

SNANCP uses the same editing keys as DCL for line editing. For example, use the ↑ or Ctrl/B keys to recall previously entered commands.

3.1.4.2 SNANCP Prompting Mode

SNANCP prompts for selected components and parameters if you do not supply them when you enter the command. For example, if you do not enter the line identification or parameters for the SET LINE command, SNANCP prompts you for the appropriate information.

Prompting does not occur when SNANCP receives input from a command file. Parameter names must be supplied in command files.

There are two ways to enter SNANCP commands. You can enter the entire command after the SNANCP prompt, or you can enter only the beginning of the command and let SNANCP prompt you for the rest of the information.

SNANCP prompts for selected components and parameters if you do not supply them when you first enter the command. These components and parameters are those that are used most often. Prompts will be either a list of valid choices for an item in a command string or the name of a parameter and a description of the value required. The following is a sample of SNANCP prompting mode for a Gateway-ST line:

```
SNANCP> SET LINE SYN-0
Line state              (ON, OFF): on
Clock mode   (EXTERNAL, INTERNAL): external
Duplex mode          (FULL, HALF): half
Protocol                (keyword): sdlc point
Receive buffers           (count): 16
Buffer size               (bytes): 521
Modem type    (NORMAL,DIAGNOSTIC): normal
Signalling     (NORMAL,NRZI,HIGH): normal
Logging enabled                  : warning
Note (quoted string, max 30 char): "Test line"
SNANCP>
```

3.1.5 Exiting From SNANCP

To exit from SNANCP, enter the EXIT command, or press Ctrl/Z after the SNANCP> prompt.

3.1.6 SNANCP Command Summary

Table 3–1 summarizes the SNANCP commands and their functions:

Table 3–1 SNANCP Commands

Command	Function	Component
CLEAR	Resets parameters to default values or removes parameters or components	ACCESS NAME CIRCUIT LINE LOGGING LU PU SERVER
EXIT	Terminates SNANCP	
HELP	Provides information on SNANCP	
LOOP	Checks communications hardware	LINE
SET	Defines or changes parameters	ACCESS NAME CIRCUIT LINE LOGGING LU PU SERVER

(continued on next page)

Table 3–1 (Cont.) SNANCP Commands

Command	Function	Component
SHOW	Displays information	ACCESS NAME CIRCUIT LINE LOGGING LU PU SERVER
SPAWN	Creates a DCL subprocess	
USE	Specifies the Gateway node that SNANCP will use	SYSTEM
ZERO	Sets counters to zero	CIRCUIT LINE LU PU SERVER

See Chapter 13 for complete descriptions of SNANCP commands.

3.2 DECnet NCP

The Gateway software recognizes a subset of DECnet NCP commands that let you manage DECnet components that communicate with the Gateway. You can use NCP commands to create, change, monitor, and remove components, such as lines and circuits on the DECnet side of the Gateway. You can also use NCP commands to run loopback tests.

NCP commands are used in the configuration procedure that is part of the installation. Part 3 of the procedure creates a file that uses NCP commands to initialize the DECnet node database on the Gateway node. See *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation* for information on the configuration command procedure.

3.2.1 Starting NCP

To start NCP, you can enter the following DCL command:

```
$ RUN SYS$SYSTEM:NCP
```

The following prompt will appear on your screen:

```
NCP>
```

Instead of entering the DCL RUN command to invoke NCP, you can create a symbol to start the utility. You can add the following in your LOGIN.COM procedure:

```
$ NCP == "$NCP"
```

Once you have invoked NCP, you can specify the node that will execute NCP commands.

3.2.2 Specifying the Gateway Node

The node where you are physically located is called your local node. By issuing network management commands at your local node, you can perform configuration, control, and monitoring functions that affect both the local node and the Gateway node. The node on which network management functions are actually performed is called the executor node. Usually, the executor node is the local node. You can enter commands at the local node, however, that specify the Gateway node as the executor node.

To specify a node other than the local node as the executor node for a series of commands, enter the following:

NCP>SET EXECUTOR NODE *gateway-node-name*

When you use this command, you can enter access control information so that you can perform privileged operations. The following command supplies a privileged username and password:

```
NCP>SET EXECUTOR NODE GWNODE USER GRAY PASSWORD MARY
```

You can also enter access control information in the following way:

```
NCP>SET EXECUTOR NODE GWNODE"GRAY MARY"::
```

To reset the executor node to the local node, enter the following:

```
NCP>CLEAR EXECUTOR NODE
```

To specify a node other than the local node as the executor node for a single command, enter the following:

NCP>TELL *gateway-node-name*"username password" SHOW KNOWN CIRCUITS

After the Gateway node provides a display of the known circuits, the local node will again be the executor node.

3.2.3 NCP Command Syntax

An NCP command can have the following parts: a command verb, a component name, one or more parameters, and one or more qualifiers; for example:

Command	Component Name	Parameter	Qualifier
SHOW	LINE *line-id*	CHARACTERISTICS	TO *file-spec*
	KNOWN LINES	COUNTERS	
	ACTIVE LINES	STATUS	
		SUMMARY	

For each command, you must supply a command verb and a component name. Optionally, you can supply one or more parameters from the parameter list and one or more qualifiers from the qualifier list.

3.2.4 Entering NCP Commands

Separate NCP verbs and parameters by spaces or tabs. The conventions used for entering NCP commands are the same as those for entering SNANCP commands (see 3.1.4). For example:

```
NCP>SET LINE ETHERNET RECEIVE BUFFERS 64
```

Use an exclamation point (!) to designate a comment line.

3.2.4.1 Command Line Editing

NCP uses OpenVMS command line editing and recall. NCP saves and allows you to access your most recently entered command. OpenVMS allows recall of up to 20 commands. Use the ↑ or Ctrl/B keys to recall a previously entered command.

3.2.4.2 NCP Prompting Mode

NCP prompts you for selected components and parameters if you do not supply them on the command line. Prompts are either a list of valid choices for an item in a command string or the name of a parameter with a description of the value required. Following is a sample of NCP prompting:

```
NCP> SET Return
(CIRCUIT, EXECUTOR, KNOWN, LINE, LOGGING, MODULE, NODE, OBJECT):
```

3.2.5 Exiting From NCP

To exit from NCP, enter the EXIT command, or press Ctrl/Z after the NCP> prompt.

3.2.6 NCP Command Summary

The NCP commands are split into two classes: commands used when the Gateway is the executor and commands used when the load host is the executor. The NCP commands executed by the Gateway are used to control the DECnet configuration. To execute these commands, set executor to the Gateway or use the NCP TELL command. Table 3–2 summarizes the NCP commands that are used when the Gateway is the executor:

Table 3–2 NCP Commands Executed by the Gateway

Command	Function	Component
CLEAR	Resets parameters to default values, removes them from the volatile database, or specifies where the NCP commands are to be executed	CIRCUIT EXECUTOR LINE LOGGING NODE
SET	Creates or modifies parameters in the volatile database or specifies where the NCP commands are to be executed	CIRCUIT EXECUTOR LINE LOGGING LOGGING EVENTS NODE
SHOW	Displays information from the volatile database	CIRCUIT EXECUTOR LINE LOGGING NODE
ZERO	Sets counters to zero	CIRCUIT LINE NODE

The commands that are executed on the load host control the ability of the load host to load the Gateway software or to perform management functions like event logging. Table 3–3 summarizes the NCP commands used when the load host is the executor:

Table 3–3 NCP Commands Executed on the Load Host

Command	Function	Component
CLEAR	Causes subsequent NCP commands to be executed by the local node	EXECUTOR NODE
CONNECT	Sets up a logical connection between the host node and the Gateway remote console interface	NODE VIA
HELP	Provides information on NCP commands	
LOAD	Loads software to a target node	NODE VIA
LOOP	Tests a particular data path	CIRCUIT EXECUTOR NODE
SET	Identifies the Gateway as the node that will execute subsequent NCP commands	EXECUTOR NODE
TELL	Identifies the executor node for the command	
TRIGGER	Starts the bootstrap mechanism of a target node to allow the node to load itself	NODE VIA

See Chapter 14 for descriptions of the NCP commands that the Gateway supports. Each command description specifies whether the command is executed by the Gateway or on the load host. See the *OpenVMS Network Control Program Manual* for information on all of the NCP commands.

3.3 SNAP

SNAP is a display utility that provides information about the communications resources a Gateway uses. SNAP can send information to video or hard-copy terminals and files. SNAP provides the following information depending on the type of terminal you use:

- For video terminals, SNAP provides a dynamic display of information about the Gateway. This display is updated approximately once per second.
- For a hard-copy terminal or file, SNAP provides a "snapshot" of current system information.

3.3.1 Starting SNAP

There are two ways to invoke SNAP on an OpenVMS system. One way is to enter a RUN command:

```
$ RUN SYS$SYSTEM:SNAP
_Node:  gateway-node-name
```

where

gateway-node-name	Is the name of the Gateway node whose resources will be displayed

Another way to invoke SNAP is to define it as a foreign command. Enter the following at the DCL level:

```
$ SNAP == "$SNAP"
$ SNAP gateway-node-name
```

If you invoke SNAP and do not include a node name, SNAP displays the Node: prompt as shown in the description of the RUN command.

You can define the logical name SNAGM$GATEWAY to be the Gateway node name. If you do so, SNAP uses the equivalence string as a default node name and does not prompt for a node name.

If your Gateway requires nonprivileged access control information to use SNAP, enter the information in the following way:

```
$ SNAP gateway-node-name"username password"::
```

3.3.2 Exiting From SNAP

To exit from SNAP, press Ctrl/Z or the SNAP command key X.

3.3.3 SNAP Command Summary

SNAP commands are bound to keys. To issue a SNAP command, simply press the indicated command key(s). Table 3–4 summarizes the SNAP command keys and their functions:

Table 3–4 SNAP Commands

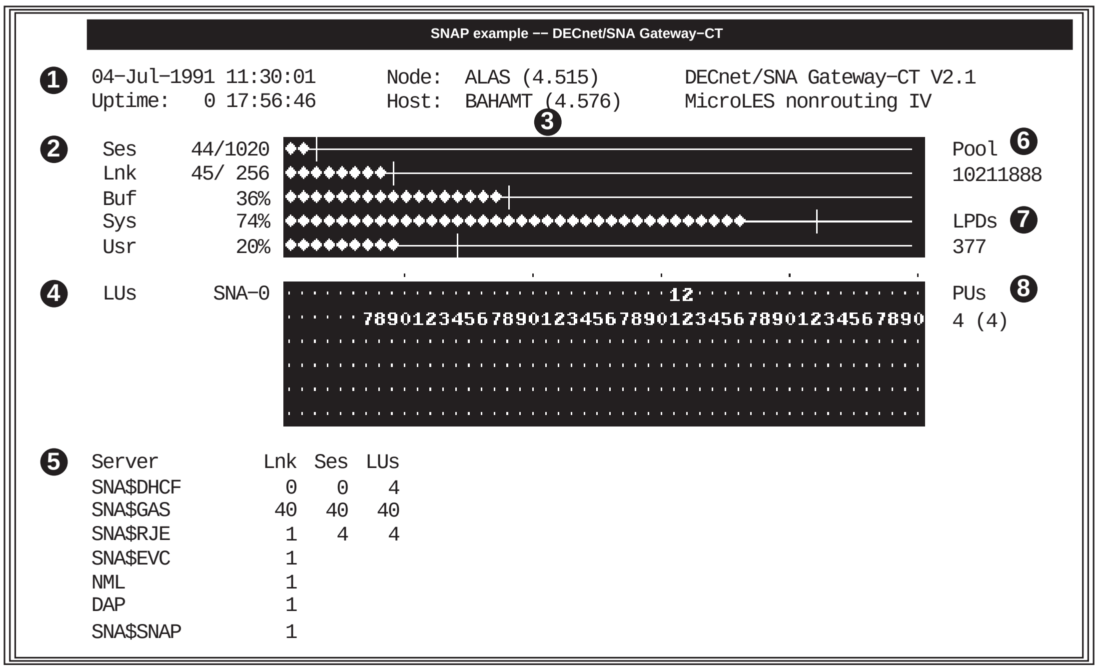
Command Keys	Function
H	Provides help on SNAP
Space or Ctrl/W	Refreshes the screen
X or Ctrl/Z	Exits from SNAP
N	Displays next active PU
Z	Resets high-water marks

When you press H, for example, SNAP displays HELP text for a short time and then returns to the display of information on system resources. If you have more PUs than can be displayed at one time, press N and SNAP advances the screen display to the next PU.

3.3.4 SNAP Display Format

Figure 3–1 shows a sample SNAP display. Numbers in black boxes mark parts of the display. These numbers correspond to the paragraphs following the figure that explain what each part of the display means. Note that the values shown in Figure 3–1 are for demonstration purposes only. Your display will show different values according to the actual resources being used.

Figure 3–1 Sample SNAP Display



LKG–5301–91R

1 The fields at the top of the display provide the following information:

- The date and time as reported by the Gateway
- The system uptime: the time elapsed since the Gateway was booted
- The Gateway node name and address (sample values: node ALAS, address 4.515)
- The host node name and address (sample values: host BAHAMT, address 4.576)
- The identification of the Gateway software (sample ID: DECnet SNA Gateway-CT V2.1 BL2)
- The type of operating system in the Gateway (MicroLES)
- The node type: nonrouting
- The DECnet phase: IV

2 This section describes the number of specific resources in use and the maximum number of resources available. For example, Figure 3–1 shows the following values:

- Sessions (Ses) — There are 44 sessions in use. The total number of LUs currently activated by ACTLU requests from the IBM host is 1020.
- Logical Links (Lnk) — There are 45 logical links active; a maximum number of 1024 logical links is allowed.
- Buffers (Buf) — The percentage of total memory in use.
- System Process (Sys) — The percentage of CPU time spent in system processes. System processes include device drivers, datalink protocols, the DECnet routing and end communications, and SNA path control.
- User Process (Usr) — The percentage of CPU time spent in user processes. User processes include servers, the SNA half-session protocol layers, and PU and LU management components.

3 The length of the bar graphically represents the percentage of resources listed in part **2** of the display in use. Each diamond shape (♦) in the bar represents a certain number of sessions, logical links, or a percentage of buffers. The specific number of sessions represented by each ♦ depends on the values returned in part **2** of the display. (DEC video terminals display a diamond shape (♦); and hard-copy terminals print an X.)

The vertical bars (|) in the bar are "high-water marks" representing the maximum number of sessions, links, and buffers that have been in use concurrently as observed by SNAP. (DEC video terminals use vertical bars; other hard-copy terminals use exclamation points.)

4 The logical unit (LU) map shows LU usage for each active physical unit associated with the Gateway.

Each dot on the bar represents an LU. When a session is in use, the dot is replaced by a digit. This digit is the low-order digit of the session address in decimal.

The marks along the top of the bar correspond to addresses 10, 20, . . . for ease of reading. For example, a 2 in the bar after the first mark indicates session 12. A 2 in the bar after the second mark indicates session 22.

SNAP displays as many PUs as will fit in the available space. If you have more PUs active than can be displayed on a single SNAP display, press N to advance the display by one PU.

5 The table displays all server processes that have LUs allocated and active logical links. A server is displayed only if it has at least one LU allocated or at least one logical link established. This table displays information for the following servers:

- SNA$GAS — SNA-ACCESS server
- SNA$GO—Gateway initializer
- SNA$RJE — DECnet SNA RJE server
- SNA$DHCF — DECnet/SNA DHCF server
- SNA$EVC — Event collector for SNAEVL
- NML — Network Management Listener
- DAP — Data access protocol PPI
- MIRROR — Logical link loopback mirror
- SNA$TRACE — Trace collector for SNATRACE
- SNA$SNAP — SNAP server
- LES$TRACE — Trace collector for NETTRACE

The following information is displayed for each server:

- Logical Links (Lnk) — Indicates the number of logical links established.
- Active Sessions (Ses) — Indicates the number of sessions currently running.
- LUs Allocated (LUs) — Indicates the number of LUs allocated to this server.

6 Pool *nnn* displays the free buffer pool figure in bytes.

7 LPDs *nnn* displays the current size of the LPD lookaside list.

8 PUs *nn* displays the count of currently active PUs and (*nn*) indicates the number of PUs defined.

For a list of the SNAP error messages, see Appendix C.

3.3.4.1 Obtaining a Copy of the SNAP Display

To obtain a hard copy of the current SNAP display, enter the following commands:

```
$ DEFINE/USER SYS$OUTPUT file-spec
$ SNAP gateway-node-name
```

where

file-spec	Is the name of the file that contains the SNAP display information.
gateway-node-name	Is the name of the Gateway node whose resources will be displayed.

4 Line Management

This chapter describes how to manage lines for the Gateway-ST and the Gateway-CT. Managing lines involves defining lines, displaying information to monitor activities on the line, and setting and clearing line parameters.

Line definitions are executed at load time. After the Gateway is loaded, you can use NCP and SNANCP commands to define additional lines and to change the values of line parameters that were defined during installation or loading. You use NCP commands to manage lines from a DECnet host to the Gateway. You use SNANCP commands to manage lines from the Gateway to the IBM network.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to take effect the next time the Gateway is reloaded, edit the following file that contains the default values for SNA components:

`SYS$COMMON:[SNA$CSV]SNAGATEWAY_`*`gateway-node-name`*`_SNA.COM`

4.1 DECnet Lines

The Gateway supports one Ethernet line connection to DECnet nodes. The Gateway is connected to the Ethernet line by a communications controller, a transceiver, and a transceiver cable. This Ethernet connection allows the Gateway load host and any other DECnet nodes (on the same network) to communicate with the Gateway.

Note

Use the following NCP commands only when the Gateway node is defined as the executor. For management of the DECnet end of the line, refer to the DECnet documentation.

On DECnet nodes, the line identification for this connection takes the following form:

dev-c

where

dev Is a device name.

c Is a decimal number (0 or a positive integer) designating the device's hardware controller.

On the Gateway node, Ethernet lines are identified as the string ETHERNET, specifying that the line is a multi-access line that uses the Ethernet protocol.

4.1.1 Defining DECnet Lines

The NCP SET LINE command defines or changes parameters for the line from the Gateway to the DECnet node while the system is running.

The following NCP commands change the value of RECEIVE BUFFERS to 64:

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>SET LINE ETHERNET RECEIVE BUFFERS 64
```

For more information on the parameters you can use with the NCP SET LINE command, and what they do, see Chapter 14.

4.1.2 Monitoring DECnet Lines

Before making changes to a line, check the current parameter values for the line. The NCP SHOW LINE command displays information about lines. By default, the information is sent to SYS$OUTPUT, which is usually your terminal. You can use the optional qualifier, TO *file-spec*, to send the information to a file.

If you enter the SHOW LINE command without any parameters, the summary display is shown:

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>SHOW KNOWN LINES

Known Line Volatile Summary as of 17-AUG-1988 16:54:20

   Line              State

  ETHERNET           on
```

The STATUS parameter gives information about the state of a line. The CHARACTERISTICS parameter gives information on parameter values.

The following is a summary display for the NCP SHOW LINE CHARACTERISTICS command:

```
NCP>SHOW LINE ETHERNET CHAR

Line Volatile Characteristics as of  5-FEB-1988 14:10:47

Line = ETHERNET

Device                    = QNA-0
Receive buffers           = 64
Controller                = normal
Protocol                  = Ethernet
Hardware address          = 08-00-2B-0A-6D-76
```

The Ethernet address associated with the Ethernet line device hardware is displayed as a read-only parameter.

For more information on the parameters you can use with the NCP SHOW LINE command, see Chapter 14.

4.1.2.1 Displaying NCP Line Counters

You can display the counters for a line by using the COUNTERS parameter:

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>SHOW LINE ETHERNET COUNTERS

Line Counters as of 17-AUG-1988 16:54:30

Line = ETHERNET
       10742  Seconds since last zeroed
        4745  Data blocks received
        4247  Multicast blocks received
           2  Receive failure, including:
                Framing error
      389349  Bytes received
      346537  Multicast bytes received
        1249  Data blocks sent
         734  Multicast blocks sent
           0  Blocks sent, multiple collision
           7  Blocks sent, single collision
           9  Blocks sent, initially deferred
       67894  Bytes sent
       41554  Multicast bytes sent
           0  Send failure
           0  Collision detect check failure
        1692  Unrecognized frame destination
           0  System buffer unavailable
           0  User buffer unavailable
```

Refer to the OpenVMS NCP reference manual for an explanation of each counter.

4.1.2.2 Zeroing NCP Line Counters

After viewing the counters, you can use the NCP ZERO LINE command to set line counters to zero.

To set the counters for all known lines to zero, enter the following commands:

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>ZERO KNOWN LINES
```

If event logging is enabled for informational messages, the counters are logged to the DECnet host when you set the counters to zero.

For additional information on the parameters you can use with the NCP ZERO LINE command, see Chapter 14.

4.1.3 Clearing DECnet Lines

After viewing information about the line and its parameters, you can change certain parameter values. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value
- SET — Provides new values for parameters

For more information on the parameters you can use with the NCP CLEAR LINE command, see Chapter 14.

4.1.4 Testing DECnet Lines

If there are problems with the lines or connections from a DECnet node to the Gateway, you can use DECnet NCP loopback tests to isolate problems. These tests check the communications software and hardware from the DECnet host to the Gateway. Run these tests if you receive DECnet error messages. See *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)* or *DECnet SNA Gateway for Synchronous Transport Problem Solving* for information on NCP loopback tests.

4.2 Gateway to IBM Lines

There are two types of interfaces to the IBM network:

- SDLC Lines — Used with the Gateway-ST. Section 4.2.1 describes SDLC lines.
- Channel-attached Lines — Used with the Gateway-CT. Section 4.2.2 describes channel-attached lines.

4.2.1 SDLC Lines

The next two sections differentiate between the line support available on the DEC MicroServer-SP (single-port) and the DEC MicroServer (four-port). Read whichever section applies to your system.

4.2.1.1 DEC MicroServer-SP Line Support

The line interface that connects the DEC MicroServer-SP to the IBM SNA network has one port for a synchronous communication line. The protocol for this line is SDLC. The DEC MicroServer-SP can handle data communications speeds of up to 19.2K bps. Table 4–1 show the line configuration available.

Table 4–1 DEC MicroServer-SP Port Speeds

Data Speeds (in bits per second)	Number of Lines	Port
Up to 19.2K bps	1	0

The line identification for the lines from the DEC MicroServer-SP to the IBM network is in the form SYN-0 where SYN is the device name and 0 designates the device's hardware controller.

4.2.1.2 DEC MicroServer Line Support

The line interface that connects the DEC DEC MicroServer to the IBM SNA network has four ports for synchronous communication lines. The protocol for these lines is SDLC. The range of line speeds is divided into three bands. In each band, a certain number of the synchronous ports are available for use. Table 4–2 shows the line configurations available.

Table 4–2 DEC MicroServer Port Speeds

Data Speeds (in bits per second)	Number of Lines	Ports
Up to 64K	4	0,1,2,3
Above 64K up to 128K	2	0 and 1
Above 128K up to 256K	1	0 only

Use the information in the preceding table in the following way:

1. Determine the maximum speed of the lines you want to use.

2. Locate the band for that speed in the table.
3. Determine the number of lines you can use.
4. Use only the ports listed for your speed band.

For example, you may want to use one line at 64K bps and another at 128K bps. In this case, the maximum data speed lies in the middle band; and so, only two ports (0 and 1) can be used.

The line identification for the lines from the DEC MicroServer to the IBM network is in the form SYN-*n*, where SYN is the device name and *n* is a decimal value from 0 to 3 that designates the device's hardware controller.

4.2.1.3 SDLC Line States

This information applies to both the DEC MicroServer-SP and the DEC MicroServer. The lines from the Gateway-ST to the IBM network have four possible states:

ON	The line has been set to the ON state.
ON–LOOPING	The SNANCP LOOP command has been issued and a loopback test is in progress.
ON–AUTO LOOPING	The IBM host is performing modem loopback tests. The state of the line will return to ON when the loopback tests have been completed.
OFF	The line is set to OFF.

A line is active when it is in any of the ON states.

The identification for lines from the Gateway-ST to the IBM network is in the form SYN-*n*, where SYN is the device name and *n* is the number of the line.

4.2.2 Channel-Attached Lines

The lines from the Gateway-CT to the IBM network have four possible states:

ON–STARTING	The line has been set to the ON state but the channel is not operational. (This state exists transiently during normal startup and may exist for longer periods if the channel fails at any time.)
ON	The line is in the ON state and the IBM channel is operational.
ON–DISABLED	The line has been set to the ON state, but the DEC ChannelServer's Channel Enable/Disable switch is in the Disable position, preventing any channel activity.
OFF	The line is set to OFF.

A line is active when it is in any of the ON states.

The identification for lines from the Gateway-CT to the IBM network is in the form CQ-0, where CQ is the device name and 0 is the number of the line. A single channel-attached line is supported by the Gateway-CT.

4.2.3 Defining Gateway to IBM Lines

The SNANCP SET LINE command defines parameters for the lines from the Gateway to the IBM network. Before using the SET LINE command to change line parameters (except COUNTER TIMER, NOTE, and LOGGING), be sure the line is in the OFF state.

For more information on the parameters you can use with the SNANCP SET LINE command, see Chapter 13.

The following example sets an SDLC line to the OFF state and changes the STATE, CLOCK, DUPLEX, RECEIVE BUFFERS, BUFFER SIZE, MODEM TYPE, SIGNAL, and LOGGING parameters, then sets the line back to the ON state:

```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> SET LINE SYN-0 STATE ON CLOCK EXT DUPLEX HALF RECEIVE 16 -
_SNANCP> BUFFER 267 MODEM NORMAL SIGNAL NORMAL LOGGING WARNING -
_SNANCP> NOTE "LINE L022B"
```

The following example sets a channel-attached line to the OFF state and changes the STATE, INTERLEAVE, and FORCED BURST parameters, and then sets the line back to the ON state:

```
SNANCP> SET LINE CQ-0 STATE OFF
SNANCP> SET LINE CQ-0 STATE ON INTERLEAVE 16 FORCED BURST DISABLED
```

4.2.4 Monitoring Gateway to IBM Lines

Before making changes to a line, check the current parameter values for the line. Use the SHOW LINE command to display information about lines. The command can be used with either SDLC lines or a channel-attached line; however, the output is different. By default, the information is sent to SYS$OUTPUT, which is usually your terminal. You can use the optional qualifier, TO *file-spec*, to send the information to a file.

For more information on the parameters you can use with the SNANCP SHOW LINE command, see Chapter 13.

Note

The following examples show the output for both an SDLC line (SYN-0) and a channel-attached line (CQ-0).

If you enter the SNANCP SHOW LINE command without any parameters, a summary display is shown.

The following example is a summary display for an SDLC line.

```
SNANCP> SHOW LINE SYN-0

Line Summary as of 21-JUN-1988 14:11:51 from THYME

   Line            State             Note

  SYN-0            on                L1228-(V.35, 56kbps, SDLC)
```

The following example is a summary display for a channel-attached line.

```
SNANCP> SHOW LINE CQ-0

Line Summary as of 21-AUG-1988 14:11:51 from STAR

   Line            State             Note

  CQ-0             on                Block multiplexer channel
```

The STATUS parameter gives information about a line's state. The CHARACTERISTICS parameter gives information on parameter values.

The following example displays the characteristics of an SDLC line.

```
SNANCP> SHOW LINE SYN-0 CHARACTERISTICS

Line Characteristics as of 21-JUN-1988 14:11:34 from THYME

Line = SYN-0

Counter timer           = 0
Receive buffers         = 64
Duplex                  = full
Protocol                = SDLC Point
Clock                   = external
Buffer size             = 1420
Signalling convention   = normal
Modem type              = normal
Logging enabled         = informational
Note                    = L1228-(V.35, 56kbps, SDLC)
```

The following example displays the characteristics of a channel-attached line.

```
SNANCP> SHOW LINE CQ-0 CHARACTERISTICS

Line Characteristics as of  5-FEB-1988 14:11:25 from STAR

Line = CQ-0
```

```
Counter timer            = 0
Receive buffers          = 64
Protocol                 = channel block
Buffer size              = 4105
Signalling convention    = normal
Forced burst             = disabled
Interleave               = 16
Logging enabled          = warning
Note                     = block multiplexer channel
```

4.2.4.1 Displaying SNANCP Line Counters

You can display the counters for a line by using the COUNTERS parameter with the SHOW LINE command. You can use the same command with SDLC lines and a channel-attached line; however, the output is different.

The SHOW LINE COUNTERS command for an SDLC line produces a display similar to the following:

```
SNANCP> SHOW LINE SYN-0 COUNTERS

Line Counters as of 21-AUG-1988 14:30:12 from THYME

Line = SYN-0

       2020 Seconds since last zeroed
     269779 Bytes received
     256689 Bytes sent
     127838 Blocks received
     127821 Blocks sent
          0 Carrier lost
          0 Process errors
          0 Receive errors
          0 Transmit errors
```

The SHOW LINE COUNTERS command for a channel-attached line produces a display similar to the following:

```
SNANCP> SHOW LINE CQ-0 COUNTERS

Line Counters as of  5-FEB-1988 14:13:53 from STAR

Line = CQ-0
```

```
   6398  Seconds since last zeroed
   3569  Commands received
   3645  Channel operations executed
 675860  Bytes received
  30554  Bytes sent
   1362  Blocks received
   1357  Blocks sent
      4  Channel events, including:
           System reset
           Selective reset
      0  Process errors
      0  Channel errors
    389  Local buffer errors, including:
           Receive buffer ring empty
     64  Buffers in use at peak
      5  Attention buffers at peak
```

For an explanation of each line counter, refer to Appendix B.

4.2.4.2 Zeroing SNANCP Line Counters

After viewing the counters, you can use the SNANCP ZERO command to set line counters to zero.

For more information on the SNANCP ZERO LINE command, see Chapter 13.

To set the counters for all known lines to zero, enter the following:

```
SNANCP> ZERO KNOWN LINES
```

If event logging is enabled for informational messages, a message is logged at the DECnet host when you set the counters to zero.

4.2.5 Clearing Gateway to IBM Lines

After viewing information about lines and their parameters, you can remove the definitions of one or more lines or change certain parameter values. To remove the definition of a line, use the SNANCP CLEAR LINE command with the ALL parameter. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value
- SET — Provides new values for parameters

Before you CLEAR any line parameter (except COUNTER TIMER, NOTE, and LOGGING), set the line state to OFF.

For more information on the SNANCP CLEAR LINE command, see Chapter 13.

If a line is already defined and you want to reset one or more line parameters to their default value, use the CLEAR LINE command with the appropriate parameters. The following SNANCP command clears the BUFFER SIZE and LOGGING parameter values:

```
SNANCP> CLEAR LINE SYN-0 BUFFER SIZE LOGGING
```

If a line is already defined and you want to specify new values for one or more line parameters, use the SET LINE command. The following SNANCP command specifies new values for BUFFER SIZE and LOGGING:

```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> SET LINE SYN-0 STATE ON BUFFER SIZE 523 -
_SNANCP>LOGGING INFORMATIONAL
```

In addition to changing the values of line parameters, you can remove one or more lines from the database. Use the CLEAR LINE command with the parameter ALL to remove a line. The following example shows how to remove the definition of a line after setting the line to OFF:

```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> CLEAR LINE SYN-0 ALL
```

To redefine a line after removing it, use the SET LINE command. When you redefine a line, the counters for that line are set to zero.

4.2.6 Testing Gateway to IBM Lines

If you have a problem with the lines or connections from the Gateway-ST to the IBM network, you can use SNANCP loopback tests to isolate the problem. These tests check the communications hardware from the Gateway to the modem at the IBM site. Run these tests if you receive SNANCP error messages. See *DECnet SNA Gateway for Synchronous Transport Problem Solving* for information on SNANCP loopback tests.

Note

SNANCP loopback tests are used only with SDLC lines.

5 Circuit Management

This chapter describes how to manage circuits for the Gateway. Communication between nodes takes place over circuits. Managing circuits involves defining circuits, displaying information to monitor activities on the circuit, and setting and clearing circuit parameters.

After installing Gateway software, you run a configuration command procedure that allows you to define circuits and circuit parameters (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). These definitions are executed at load time. After the Gateway is loaded, you can use NCP and SNANCP commands to define additional circuits and to change the values of circuit parameters that were defined during installation or loading. You use NCP commands to manage circuits from a DECnet host to the Gateway. You use SNANCP commands to manage circuits from the Gateway to the IBM network.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to be used the next time the Gateway is reloaded, edit the following file that contains the default values for Gateway components:

`SYS$COMMON:[SNA$CSV]SNAGATEWAY_gateway-node-name_SNA.COM`

5.1 DECnet Circuits

The circuit from the DECnet host to the Gateway is an Ethernet circuit. Ethernet circuits provide a multi-access connection between a number of nodes on the same broadcast circuit. Every node must have a unique node identification: an Ethernet physical address. Ethernet circuits use the Ethernet protocol.

Note

Use the following NCP commands only when the Gateway node is defined as the executor. For management of the DECnet end of the circuit, refer to the *VMS Network Control Program Manual*.

On DECnet nodes, circuit identification takes the following form:

dev-c

where

dev	A device name.
c	A decimal number (0 or a positive integer) designating the device's hardware controller.

On the Gateway node, the Ethernet circuit is identified by the string ETHERNET.

5.1.1 Defining DECnet Circuits

The NCP SET CIRCUIT command defines or changes circuits and parameters for the circuits from the DECnet host to the Gateway.

The following commands set the ETHERNET circuit HELLO TIMER to 10 seconds.

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>SET CIRCUIT ETHERNET HELLO TIMER 10
```

For more information on the NCP SET CIRCUIT command, see Chapter 14.

5.1.2 Monitoring DECnet Circuits

Before making changes to a circuit, check the current parameter values. The NCP SHOW CIRCUIT command displays information about circuits. By default, the information is sent to SYS$OUTPUT, which is generally your terminal. You can use the optional qualifier, TO *file-spec*, to send the information to a file.

If you enter the SHOW CIRCUIT command without any parameters, a SUMMARY display is shown; for example:

```
NCP>SET EXECUTOR NODE SNAGWY
NCP>SHOW KNOWN CIRCUITS

Known Circuit Volatile Summary as of 18-JAN-1988 15:13:11
```

```
  Circuit          State                        Loopback     Adjacent
                                                  Name         Node
 ETHERNET          on                                          4.31
```

The STATUS parameter gives information about the state of the circuit. The CHARACTERISTICS parameter gives information about parameter values:

```
NCP>SHOW KNOWN CIRCUITS CHARACTERISTICS

Known Circuit Volatile Characteristics as of 18-JAN-1988 15:13:20

Circuit = ETHERNET

State                    = on
Adjacent node            =  4.31
Designated router        =  4.31
Block size               = 1498
Hello timer              = 15
Listen timer             = 45
Line                     = ETHERNET
Type                     = Ethernet
```

For more information on the NCP SHOW CIRCUIT command, see Chapter 14.

5.1.2.1 Displaying NCP Circuit Counters

You can display the counters for a circuit by using the COUNTERS parameter with the SHOW CIRCUIT command; for example:

```
NCP>SET EXECUTOR NODE SNAGWY USER SYSTEM PASSWORD SECRET
NCP>SHOW CIRCUIT ETHERNET COUNTERS

Circuit Counters as of 18-JAN-1988 15:22:54

Circuit = ETHERNET

        65534  Seconds since last zeroed
          415  Terminating packets received
          385  Originating packets sent
            0  Terminating congestion loss
            0  Corruption loss
            0  Transit packets received
            0  Transit packets sent
            0  Transit congestion loss
            0  Circuit down
            0  Initialization failure
            0  User buffer unavailable
     12559257  Bytes received
       508870  Bytes sent
        77226  Data blocks received
        12774  Data blocks sent
```

For an explanation of each counter, refer to the *OpenVMS Network Control Program Manual*.

5.1.2.2 Zeroing NCP Circuit Counters

After viewing the counters, you can use the NCP ZERO CIRCUIT command to set the circuit counters to zero.

To set the counters for the ETHERNET circuit to zero, enter the following commands:

```
NCP>SET EXECUTOR SNAGWY USER SYSTEM PASSWORD SECRET
NCP>ZERO CIRCUIT ETHERNET
```

If event logging is enabled for informational messages, a message is logged when you set the counters to zero. For more information on the NCP ZERO CIRCUIT command, see Chapter 14.

5.1.3 Clearing DECnet Circuits

After viewing information about circuits and their parameters, you can remove one or more parameters or change certain parameter values. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value
- SET — Provides new values for parameters

The following command resets the COUNTER TIMER parameter to the default value:

```
NCP>SET EXECUTOR SNAGWY USER SYSTEM PASSWORD SECRET
NCP>CLEAR CIRCUIT ETHERNET COUNTER TIMER
```

If you want to specify new values for one or more circuit parameters, use the SET CIRCUIT command with the appropriate parameters. The following command changes the value of the COUNTER TIMER parameter to 25:

```
NCP>SET CIRCUIT ETHERNET COUNTER TIMER 25
```

For more information about the NCP CLEAR CIRCUIT command, see Chapter 14.

5.2 Gateway to IBM Circuits

Circuits provide the protocol layer between a PU and a line. You can use either of the following types of circuits:

- SDLC circuits — Used with the Gateway-ST. Each SDLC line supports a single circuit.

- Channel-attached circuits — Used with the Gateway-CT. Each channel-attached line can support several circuits. Each channel-attached circuit corresponds to one address on the channel.

5.2.1 SDLC Circuits

An SDLC circuit can be in any of the following states:

OFF	The circuit is set to OFF.
ON–STARTING	The circuit has been set ON, and the Gateway-ST is waiting for the IBM host to establish contact.
ON	The IBM host is communicating with the Gateway-ST.
ON–TIMED OUT	The Gateway-ST has lost communication with the IBM host because no frames have been seen for a specified length of time.
ON–DISCONNECTED	The IBM host has ended the communication with the Gateway-ST.
ON–AUTOLOOPING	The IBM host is performing modem loopback tests. The state of the circuit will return to ON when the loopback tests have been completed.

The circuit identification specifies the protocol of the circuit. The circuit identification for SDLC circuits is SDLC-*nnn*, where *nnn* is the number of the circuit, for example, SDLC-0.

5.2.2 Channel-Attached Circuits

A channel-attached circuit can be in any of the following states:

OFF	The circuit is set to OFF.
ON–STARTING	The circuit has been set ON, and the Gateway-CT is waiting for the IBM host to establish contact.
ON	The IBM host is communicating with the Gateway-CT.
ON–DISCONNECTED	The IBM host has ended communication with the Gateway-CT.

The circuit identification specifies the protocol of the circuit. The circuit identification for channel-attached circuits is CHAN-*nnn*, where *nnn* is the number of the circuit; for example, CHAN-2.

5.2.3 Defining Gateway to IBM Circuits

The SNANCP SET CIRCUIT command defines circuits and parameters for the circuits from the Gateway to the IBM network. Before using the SET CIRCUIT command to change circuit parameters (except COUNTER TIMER, NOTE, and LOGGING), be sure the circuit is in the OFF state.

The following example sets a SDLC circuit to the OFF state, changes the values for DUPLEX, LINE, and NOTE, and returns the circuit to the ON state:

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> SET CIRCUIT SDLC-0 STATE ON DUPLEX HALF LINE SYN-0 -
SNANCP> NOTE "IBM Line L022B"
```

For more information on the SNANCP SET CIRCUIT command, see Chapter 13.

5.2.4 Monitoring Gateway to IBM Circuits

Before making changes to a circuit, check the current parameter values. The SNANCP SHOW CIRCUIT command displays information about circuits. The command can be used with either SDLC or channel-attached circuits; however, the output is different. By default, the information is sent to SYS$OUTPUT which is generally your terminal. You can use the optional qualifier TO *file-spec* to send the information to a file.

If you enter the SHOW CIRCUIT command without any parameters, a SUMMARY display is shown.

The circuit summary for an SDLC circuit is as follows:

```
SNANCP> SHOW KNOWN CIRCUITS

Known Circuit Summary as of 29-AUG-1988 18:11:26 from THYME

   Circuit          State                         Note

  SDLC-0            on                            Line L0226B on IBM
```

The circuit summary for a channel-attached circuit is as follows:

```
SNANCP> SHOW KNOWN CIRCUITS

Known Circuit Summary as of 29-AUG-1988 18:11:26 from STAR

   Circuit          State                         Note

   CHAN-1           on                            Majnode H01LE
```

The STATUS parameter provides information about the state of the circuit. The CHARACTERISTICS parameter gives information about parameter values.

The following example displays the characteristics of all known SDLC circuits.

```
SNANCP> SHOW KNOWN CIRCUITS CHARACTERISTICS

Known Circuit Characteristics as of 29-AUG-1988 18:11:29 from THYME

Circuit = SDLC-0

Counter timer             = 0
Line                      = SYN-0
Logging enabled           = warning
Duplex                    = full
Response mode             = normal
Station address           = 40
Station ID                = 00013790
Idle timer                = 30
Note                      = Line L02268B on IBM
```

The following example displays the characteristics of all known channel-attached circuits.

```
SNANCP> SHOW KNOWN CIRCUITS CHARACTERISTICS

Known Circuit Characteristics as of 29-AUG-1988 18:11:29 from STAR

Circuit = CHAN-1

Counter timer             = 0
Line                      = CQ-0
Logging enabled           = warning
Channel address           = 090
Note                      = Majnode H01LE
```

For more information on the SNANCP SHOW CIRCUIT command, see Chapter 13.

5.2.4.1 Displaying SNANCP Circuit Counters

You can display the counters for a circuit by using the COUNTERS parameter with the SHOW CIRCUIT command. You can use the same command with SDLC and channel-attached circuits; however, the output is different.

The SHOW CIRCUIT COUNTERS command for an SDLC circuit displays the following:

```
SNANCP> SHOW CIRCUIT SDLC-0 COUNTERS

Circuit Counters as of 21-AUG-1988 08:28:45 from THYME

Circuit = SDLC-0

       14729  Seconds since last zeroed
       14361  Bytes received
        1530  Bytes sent
         130  Data blocks received
         117  Data blocks sent
           0  Data errors inbound
           0  Data errors outbound
           0  Remote buffer errors
           0  Local buffer errors
           0  Remote station errors
           0  Local station errors
           0  Test frames received
           0  Test frames received, bad CRC
      931628  Polls received
           0  Frame retransmits
```

The SHOW CIRCUIT COUNTERS command for a channel-attached circuit displays the following:

```
Known Circuit Counters as of 19-AUG-1988 08:28:45 from THYME

Circuit = CHAN-1

        2236  Seconds since last zeroed
      328830  Bytes received
       12925  Bytes sent
         525  Data blocks received
         526  Data blocks sent
           4  Channel process errors, including:
                Data length error
                Connect rejected, invalid parameters
         231  Attentions sent
         312  Attentions sent on read
         307  Write channel programs
         531  Read channel programs
```

For an explanation of what each counter means, refer to Appendix B.

5.2.4.2 Zeroing SNANCP Circuit Counters

After viewing the counters, you can use the SNANCP ZERO CIRCUIT command to set the circuit counters to zero.

To set the counters for circuit SDLC-0 to zero, enter the following command:

```
SNANCP> ZERO CIRCUIT SDLC-0
```

If event logging is enabled for informational messages, a message is logged at the DECnet host when you set the counters to zero.

For more information on the SNANCP ZERO CIRCUIT command, see Chapter 13.

5.2.5 Clearing Gateway to IBM Circuits

After viewing information about circuits and their parameters, you may want to remove the definitions of one or more circuits or change certain parameter values. To remove the definition of one or more circuits, use the CLEAR CIRCUIT command with the parameter ALL. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value.
- SET — Provides new values for parameters.

Before you CLEAR any circuit parameter (except COUNTER TIMER, NOTE, and LOGGING), set the circuit state to OFF.

For more information on the SNANCP CLEAR CIRCUIT command, see Chapter 13.

The following command resets the COUNTER TIMER parameter to the default value:

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> CLEAR CIRCUIT SDLC-0 COUNTER TIMER
```

If a circuit is already defined and you want to specify new values for one or more circuit parameters, use the SET CIRCUIT command with the appropriate parameters. The following command allocates a new line to the circuit:

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> SET CIRCUIT SDLC-0 DUPLEX HALF
```

In addition to changing the values of circuit parameters, you can remove one or more circuits from the database. Use the CLEAR CIRCUIT command with the parameter ALL to remove a circuit.

The following example removes the definition of the circuit SDLC-0 after setting the circuit to OFF:

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> CLEAR CIRCUIT SDLC-0 ALL
```

To redefine a circuit after removing it, use the SET CIRCUIT command. When you redefine a circuit, the counters for that circuit are set to zero.

6

PU Management

This chapter describes how to manage physical units (PUs) for the Gateway. Managing PUs involves defining PUs, displaying information to monitor the activities of a PU and its associated logical units (LUs), and setting and clearing PU parameters.

Note

The SNANCP PU commands described in this chapter can affect the list of LUs associated with a particular PU. For a description of LUs and the SNANCP commands that define and clear parameters for LUs, see Chapter 7.

After installing Gateway software, you run a configuration command procedure that defines PUs and PU parameters (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). These definitions are executed at load time. After the Gateway is loaded, you can use SNANCP commands to define additional PUs and to change the values of PU parameters that were defined during installation or loading.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to be used the next time the Gateway is reloaded, edit the following file that contains the default values for SNA components:

SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_SNA.COM

6.1 Gateway PUs

An IBM system perceives all communications with an access node in terms of logical connections called sessions. These sessions are similar to DECnet logical links. The software components linked together in a session—for example, an application in a DECnet node and an application in an IBM host – are identified as logical units (LUs). Communication between LUs is accomplished by means of Gateway software components called physical units (PUs). In DECnet SNA applications, each PU has a one-to-one correspondence with a particular circuit.

A PU can be ACTIVE or INACTIVE. ACTIVE means an SNA SSCP-PU session is in progress. INACTIVE means there is no SSCP-PU session.

A PU identification is in the form SNA-*nnn*, where *nnn* is a decimal number identifying the PU; for example, SNA-0.

6.2 Defining PUs

The SNANCP SET PU command defines PUs and their associated parameters. The SET PU command also defines associated LUs for a PU. Before using the SET PU command to change PU parameters (except COUNTER TIMER, NOTE, and LOGGING), be sure that the PU is inactive. Unlike CIRCUIT state and LINE state, you cannot set the PU state directly. To make a PU inactive, you must set the corresponding circuit to OFF.

The following command changes the values of CIRCUIT and NOTE for PU SNA-0:

```
SNANCP> SET CIRCUIT SDLC-0 OFF
SNANCP> SET PU SNA-0 CIRCUIT SDLC-1 NOTE "IBM PU P022B"
```

For more information on the SNANCP SET PU command, see Chapter 13.

6.3 Monitoring PUs

Before making a change to a PU, check the current parameter values for the PU. The SNANCP SHOW PU command displays information about PUs. By default, the information is sent to SYS$OUTPUT, which is usually your terminal. You can use the optional qualifier TO *file-spec* to send the information to a file.

For more information on the SNANCP SHOW PU command, see Chapter 13.

If you enter the SHOW PU command without any parameters, a summary display (PU identification, state, and note text) is shown:

```
SNANCP> SHOW PU SNA-0

PU Summary as of 21-APR-1988 13:21:88 from THYME

  PU                State                     Note

  SNA-0             active                    IBM PU P022B
```

The STATUS parameter gives information about the state of a PU.

The CHARACTERISTICS parameter gives information about PU parameters:

```
SNANCP> SHOW KNOWN PUS CHARACTERISTICS

Known PU Characteristics as of 24-AUG-1988 15:32:39 from THYME

PU = SNA-0

Counter timer            = 0
Logging enabled          = warning
LU list                  = 1-255
Segment size             = 9212
Query threshold          = 0
Circuit                  = CHAN-0
Note                     = IBM PU P022B
```

6.3.1 Displaying PU Counters

You can display the counters for a PU by using the COUNTERS parameter with SHOW PU; for example:

```
SNANCP> SHOW PU SNA-0 COUNTERS

PU Counters as of 24-AUG-1988 15:32:30 from THYME

PU = SNA-0

       11252  Seconds since last zeroed
           0  Valid REQMS received
           0  Invalid REQMS received
           0  REQMS receive threshold exceeded
           2  ACTPUs received, including:
                ACTPUs (ERP)
           0  Negative responses
           1  DACTPUs received
         520  Total ACTLUs received
          60  Total DACTLUs received
         255  Active LUs
```

Notice that the SHOW PU COUNTERS command displays the number of active LUs for the PU you specify. See Chapter 7 for information on displaying other information about LUs.

6.3.2 Zeroing PU Counters

After viewing the counters, you can use the SNANCP ZERO PU command to set the PU counters to zero.

For more information on the SNANCP ZERO PU command, see Chapter 13.

To set the counters for PU SNA-0 to zero, enter the following command:

```
SNANCP> ZERO PU SNA-0
```

If event logging is enabled for informational messages, a message is logged at the load host when you set the counters to zero.

6.4 Clearing PUs

After viewing information about PUs and their parameters, you can remove the definitions of one or more PUs or change certain parameter values. To remove the definition of a PU, use the SNANCP CLEAR PU command with the ALL parameter. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value.
- SET — Provides new values for parameters.

For more information on the SNANCP CLEAR PU command, see Chapter 13.

A PU must be inactive before you use the CLEAR PU command to remove the PU or its associated LU list from the Gateway database. To make a PU inactive, set the state of the corresponding circuit to OFF.

The following command resets the COUNTER TIMER and QUERY THRESHOLD parameters to their default values:

```
SNANCP> CLEAR PU SNA-0 COUNTER TIMER THRESHOLD
```

The following command specifies new values for the COUNTER TIMER and QUERY THRESHOLD parameters:

```
SNANCP> SET PU SNA-0 COUNTER TIMER 60 THRESHOLD 60
```

To clear SSCP IDs for a PU, include the SSCP ID parameter keyword in the CLEAR PU command. When you do this, you clear the entire list of SSCP IDs. To remove individual SSCP IDs, you must first clear the existing list and then issue a SET PU *pu-id* SSCP ID *sscp-id list* to restore the SSCP IDs that you want to keep. The following command removes all SSCP IDs and the LU list for the PU SNA-0.

```
SNANCP> CLEAR PU SNA-0 SSCP ID LU LIST
```

In addition to changing the value of PU parameters, you can remove one or more PUs from the Gateway database. Use the CLEAR PU command with the ALL parameter to remove a PU. The following example removes the definition of PU SNA-0:

```
SNANCP> SET CIRCUIT SDLC-0 OFF
SNANCP> CLEAR PU SNA-0 ALL
```

The next chapter explains how to obtain more information about the LUs defined for a PU.

7 LU Management

This chapter describes how to manage logical units (LUs) for the Gateway system. Managing LUs involves defining parameters for LUs, displaying information to monitor the activities of LUs, and clearing LU parameters.

Note

To create or clear LU definitions, use the SNANCP PU commands described in Chapter 6. The SNANCP LU commands described in this chapter affect only LU parameters, not the definition of an LU itself.

After installing Gateway software, you run a configuration command procedure that allows you to define physical units (PUs) and any associated LUs (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). These definitions are executed at load time. After the Gateway is loaded, you can use SNANCP commands to define additional PUs and LUs and to change the values of PU and LU parameters that were defined during installation or loading.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to be used the next time the Gateway is reloaded, edit the following file that contains the default values for SNA components:

`SYS$COMMON:[SNA$CSV]SNAGATEWAY_`*`gateway-node-name`*`_SNA.COM`

7.1 Gateway LUs

In DECnet to SNA communications, the software components that are linked together in a session—for example, an application on an access node and an application on an IBM host—are identified as logical units (LUs). Each LU can be used for only one LU–LU session at a time. A certain number of LUs are defined in the SNA host for use by different types of applications.

When a session is established, the IBM host designates one LU as the primary logical unit (PLU) and one LU as the secondary logical unit (SLU). In communications involving the Gateway, the IBM application is always the PLU and the DECnet SNA access routine is always the SLU.

An LU can be in any one of the following states:

INACTIVE	This Gateway LU does not have an active session with an SSCP.
INACTIVE–LISTENING	The Gateway is waiting for IBM to initiate a session. This state is the same as INACTIVE except that a server is connected and listening for a BIND on the LU–LU session.
PENDING ACTIVE	The Gateway is in the process of setting up the LU–SSCP session. The Gateway is waiting to connect to VTAM.
PENDING ACTIVE–LISTENING	The Gateway is waiting to connect to the IBM application. This state is the same as PENDING–ACTIVE except that a server is connected and is listening for a BIND.
ACTIVE	An LU–SSCP session is active.
ACTIVE–LISTENING	Same as ACTIVE except that a server is connected and listening on the LU (it is listening for a BIND on the LU–LU session).
ACTIVE–IN-SESSION	LU-LU session is established.
ACTIVE–BIND REQUESTED	An LU–SSCP session is active and a server has requested an LU–LU session with an IBM application. No BIND message has been received.
ACTIVE–BIND IN PROGRESS	An LU–LU session with an IBM application has been requested and a BIND from the IBM application is being processed.

ACTIVE–UNBIND–REQUESTED	An LU-SSCP session is active and a server has requested the SSCP to end the LU-LU session with an IBM application. No UNBIND message has been received yet.
ACTIVE–UNBIND–IN–PROGRESS	An UNBIND message has been received for an LU-LU session and in being processed.
ACTIVE–DISCONNECTING	The LU-LU session has ended.
PENDING INACTIVE	The LU–SSCP session is in the process of being terminated.
PENDING INACTIVE–LISTENING	Same as PENDING–INACTIVE except that a server is listening on the LU.

The form of an LU identifier is *pu-id.nnn*

where

pu-id	Is in the form SNA-*nnn*; for example, SNA-0.
nnn	Is an LU number (from 1 to 255) within the range of LUs already defined for the PU.

An example of an LU identifier is SNA-0.1.

7.2 Defining LUs

The SNANCP SET PU command defines the LUs that are associated with a particular PU. The SNANCP SET LU command defines parameters for a particular LU. After you have defined an LU with the SET PU command, you can do the following with the SET LU command:

- Control access to the LU with an authorization entry
- Specify the logging level for the LU
- Add note text for the LU

For more information on the SNANCP SET LU command, see Chapter 13.

7.2.1 Defining Authorization Entries with Component Qualifiers

Authorization entries let you limit access to Gateway functions on an LU basis. You can specify that only certain nodes, users, and/or terminals can have access to specific LUs. Refer to the documentation for the individual access routine products to see if the product you are using supports LU authorization.

To control access to individual LUs, use one of the following component qualifiers:

AUTHORIZATION *auth-id*—Defines or changes a specific authorization entry for the specified LU(s).

KNOWN AUTHORIZATIONS—Defines or changes all known authorization entries for the specified LU(s).

When you attempt to use an LU that has an authorization entry, the Gateway tests your access information against each of the fields in the specified authorization entries. For access to be granted, the access information must match all four fields in a particular authorization entry. If the information does not match any one of the existing authorization entries, access is denied.

If you create authorization entries or change authorization information for an LU that is currently in use, the authorization information does not affect the current session. The Gateway checks authorization information the next time it attempts to allocate the LU.

Using KNOWN LUs is both faster and uses less memory than issuing a separate command for each LU.

Note that some products do not provide for a username, terminal name, or password.

By default, there is no authorization information. Without authorization information for an LU, the LU is not secure; anyone is allowed to use it.

The following command creates an authorization entry for SNA-0.1:

```
SNANCP> SET LU SNA-0.1 AUTHORIZATION NEW_AUTH NODE BRETTUSER SYSTEM -
_SNANCP> PASSWORD MYPASS
```

For more information on using the SNANCP SET LU command to define authorization entries, see Chapter 13.

The following example applies identical authorization to multiple LUs. First, a PU is created with 128 LUs. An authorization entry is created for each LU. Then, the LU list is expanded to 255 LUs. Note that LUs 129 through 255 have no authorization entry because they did not exist when the SET KNOWN LUS command was issued.

```
SNANCP> SET PU SNA-0 LU LIST 1-128
SNANCP> SET KNOWN LUS AUTH KEEPOUT NODE ALASKA USER POLARBEAR
SNANCP> SET PU SNA-0 LU LIST 1-255
```

7.2.2 Defining LU Parameters

Use any of the following parameters with the SET LU command:

LOGGING [ENABLED]—Specifies which of the following levels of event messages is recorded:

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

The event level you specify sets the minimum level to be logged; all levels of greater severity will also be logged. For example, if you set logging to INFORMATIONAL, messages at all other levels will also be logged. Fatal event messages are always logged. For information on event messages, severity levels, and logging commands, see Chapter 10.

The initial value for the logging parameter is taken from the PU definition when the SET PU LU LIST command is executed. However, the logging level you specify with SET LU *lu-component* LOGGING replaces the logging level previously defined.

NOTE—Specifies additional information (no more than 30 characters) that are part of the SHOW SUMMARY display. Specify the text of your note within quotation marks.

The following command changes the values of the LOGGING and NOTE parameters for LU SNA-0.1:

```
SNANCP> SET LU SNA-0.1 LOGGING WARNING NOTE "IBM LUS L022B01-19"
```

7.3 Monitoring LUs

Before making changes to LU parameters, check the current parameter values for the LU. The SHOW LU command displays information about LU parameters. By default, the information is sent to SYS$OUTPUT, which is generally your terminal. You can use the optional qualifier TO *file-spec* to send the information to a file.

For more information on the SNANCP SHOW LU command, see Chapter 13.

If you enter the SHOW LU command without any parameters, a summary display (LU identification, state, and note text) is shown:

```
SNANCP> SHOW LU SNA-0.1

LU Summary as of 21-APR-1988 13:21:88 from THYME

  LU              State                     Note
  SNA-0.1         active
```

The CHARACTERISTICS parameter gives information about LU parameters. The following command displays information about LU SNA-0.1:

```
SNANCP> SHOW LU SNA-0.1 CHARACTERISTICS

LU Characteristics as of 16-MAR-1988 20:51:33 from THYME

LU = SNA-0.1

Logging enabled          = fatal
```

If an LU is defined to have outbound session allocation, the Outbound Server value is the name of the server for which that service is enabled. See Chapter 9 for more information on how to allocate LUs for outbound connections (connections initiated by a PLU) on an IBM system to a DECnet object.

The STATUS parameter displays the LU identification, status information, and any of the following BIND parameters (if the LU is in any of the active states):

Parameter	Format
LU type	*n*
Primary LU	text
RU size (in)	*nnnn*
RU size (out)	*nnnn*
TS profile	*nn*
FM profile	*nn*
Last receive check	*xxxx*
Last transmit check	*xxxx*
Secondary send pacing	*nn*
Secondary receive pacing	*nn*
Logon mode	text
User data	text

Note

The user data BIND parameter is displayed only if you use privileged access control.

The Last receive check shows the IBM status code for the last error that occurred while the Gateway was receiving information. The Last transmit check shows the IBM status code for the last error that occurred while the Gateway was transmitting information. Look up the sense code in the IBM *Systems Network Architecture Formats* manual. The Last receive check and Last transmit check parameters are suppressed when zero. Other BIND parameters are suppressed when blank.

Following is a sample STATUS display:

```
SNANCP> SHOW LU SNA-0.1 STATUS

LU Status as of 11-AUG-1988 16:40:00 from THYME

LU = SNA-0.1

State                 = active
Last transmit check   = 0821
```

7.3.1 Displaying LU Authorization Information

The authorization entries for an LU can be shown with the SNANCP SHOW LU command.

If you do not specify CHARACTERISTICS or SUMMARY, the summary display is the default.

To execute this command, you must have entered a privileged username and password in the SNANCP USE SYSTEM command. (For more information on the privileged username and password, see Chapter 2). The following command displays output for a privileged user only:

```
SNANCP> SHOW LU SNA-0.1 KNOWN AUTHORIZATION CHARACTERISTICS

LU Known Authorization Characteristics as of 7-JAN-1988 14:56:51 from THYME

LU = SNA-0.1

Authorization ID         = DONNA
Node                     = BRETT
User                     = SYSTEM
Terminal                 = *
Password                 = password
```

For security reasons the actual password is not displayed. Instead, the word "password" indicates that an authorization password has been set.

For more information on using the SNANCP SHOW LU command to display LU authorization information, see Chapter 13.

7.3.2 Displaying LU Counters

You can display the counters for an LU by using the COUNTERS parameter with the SHOW LU command; for example:

```
SNANCP> SHOW LU SNA-0.1 COUNTERS

LU Counters as of 11-AUG-1988 16:55:13 from THYME

LU = SNA-0.1

     332468 Seconds since last zeroed
          2 ACTLUs received, including:
               ACTLUs (ERP)
          0 BINDs received
          0 BINDs accepted
          0 BINDs rejected
          1 DACTLUs received
          0 Resource errors
          0 Session establishment failures
```

7.3.3 Zeroing LU Counters

After viewing the counters, you can use the SNANCP ZERO LU command to set the LU counters to zero.

For example, to set the counters for LU SNA-0.8 to zero, enter the following command:

```
SNANCP> ZERO LU SNA-0.8 COUNTERS
```

If event logging is enabled for informational messages, a message is logged at the DECnet host when you set the counters to zero.

For more information on using the SNANCP ZERO LU command to zero LU counters, see Chapter 13.

7.4 Clearing LU Parameters

After viewing information about LU parameters, you can remove certain parameters or change their values. To change parameter values, use the CLEAR or SET commands:

- CLEAR — Resets parameters to their default value.
- SET — Provides new values for parameters.

The following parameters can be used with the CLEAR LU command:

The following command resets the NOTE parameter to its default value (which is none):

```
SNANCP> CLEAR LU SNA-0.1 NOTE
```

The following command provides new text for the NOTE parameter:

```
SNANCP> SET LU SNA-0.1 NOTE "IBM LU L022B01"
```

Clearing LU AUTHORIZATION Parameters

The CLEAR LU AUTHORIZATION command removes one or more of the individual parameters for an authorization entry. You cannot clear authorization information on the same command line with the ALL, LOGGING, or NOTE parameters.

For more information on the SNANCP CLEAR LU AUTHORIZATION command, see Chapter 13.

The following command removes the NODE and USER information from the authorization entry NEW_AUTH:

```
SNANCP> CLEAR LU SNA-0.1 AUTHORIZATION NEW_AUTH NODE USER
```

You must use a separate CLEAR LU command to remove authorization entries themselves. Use the keyword ALL to remove the entire authorization entry from the database. The following command removes the authorization entry NEW_AUTH:

```
SNANCP> CLEAR LU SNA-0.1 AUTHORIZATION NEW_AUTH ALL
```

For more information on using the SNANCP CLEAR LU command to clear LU parameters, see Chapter 13.

8 Access Name Management

This chapter describes how to manage access names for the Gateway. Managing access names involves defining, displaying information, and changing parameters for access names.

Access names are optional components (unlike lines, circuits, PUs, and LUs) that you can define to represent IBM application access information. There are no default access names and no default access name parameters. After installing Gateway software, you run a configuration command procedure that lets you create access names (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). If you choose to define access names, these definitions are executed at load time. After the Gateway is loaded, you can use SNANCP commands to define additional access names and to change the values of access name parameters that were defined during installation or loading.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to be used the next time the Gateway is reloaded, edit the following file that contains the default values for SNA components:

`SYS$COMMON:[SNA$CSV]SNAGATEWAY_`*`gateway-node-name`*`_SNA.COM`

8.1 Gateway Access Names

An access name represents a list of parameters required for access to an application on an IBM host. Access names are similar to OpenVMS logical names; they are short symbolic references to other longer entities.

You do not have to use access names. Users of the DECnet SNA OpenVMS access routines can specify access information each time they use the individual access routine products. However, access names provide a simple way of passing several types of access information to the IBM system. Access names are optional with the DECnet SNA OpenVMS access routines except in the following cases:

- RJE requires access names.
- DHCF does not use access names. Refer to Chapter 9 for information on how to specify parameters for DHCF.

Refer to the user documentation for the individual DECnet SNA access routines for specific information on using access names with a particular product.

8.2 Defining Access Names

The SNANCP SET ACCESS NAME command defines access names. An access name is a string of 1 to 16 alphanumeric characters; for example, NCCF.

For more information on the SNANCP SET ACCESS NAME command, see Chapter 13.

Note

Check with the IBM systems programmers at your site to determine which values you must specify for the access-name parameters. Some IBM applications do not require all of the access name parameters.

Example

The following commands define access names for two IBM applications, CICS and TSO:

```
SNANCP> SET ACCESS NAME CICS PU SNA-0 LU LIST 1-4 APPLICATION CICS -
_SNANCP> NOTE "access to CICS"
SNANCP> SET ACCESS NAME TSO PU SNA-0 LU LIST 5-8 APPLICATION TSO -
_SNANCP> NOTE "access to TSO"
```

8.3 Displaying Access Names

Before making changes to access name parameters, you can check the current parameter values. The SHOW ACCESS NAME command displays information about access names. By default, the information is sent to SYS$OUTPUT, which is usually your terminal. You can use the optional qualifier TO *file-spec* to send the information to a file.

For more information on the SNANCP SHOW ACCESS NAME command, see Chapter 13.

If you enter the SHOW ACCESS NAME command without a parameter, a summary display is shown:

```
SNANCP> SHOW KNOWN ACCESS NAMES

Known Access name Summary as of 29-AUG-1988 19:25:17 from THYME

  Access name                              Note
  CICS                              PU 122c1, LU Type 2 sessions
  CICSL62
  DISOSS
  DISOSS31
  IMS
  NCCF                              PU 11140, LU Type 2 sessions
  RJE0512
  RJESMALL
  TSO
```

When using the SHOW ACCESS NAME CHARACTERISTICS command, a value for the DATA parameter is displayed only if you enter a privileged username and password in the SNANCP USE SYSTEM command. If there is a value for the DATA parameter and you have not entered a privileged username and password, you see the string *SET*, which indicates that there is a value. For more information on privileged and nonprivileged usernames and passwords, see Chapter 2.

The following command shows parameter values for the access name TSO:

```
SNANCP> SHOW ACCESS NAME TSO CHARACTERISTICS

Access name Characteristics as of 25-AUG-1988 16:38:28 from THYME

Access name = TSO

PU                        = SNA-0
LU list                   = 1-64
PU                        = SNA-1
LU list                   = 1-255
Logon mode                = TSOE2
Application               = TSO
```

8.4 Clearing Access Names

The SNANCP CLEAR ACCESS NAME command removes an access name definition or one or more access name parameters from the Gateway database. There are no default access names and no default access-name parameters.

For more information on the SNANCP CLEAR ACCESS NAME command, see Chapter 13.

The following command removes the PU/LU-list pairs that have been defined for TSO:

```
SNANCP> CLEAR ACCESS NAME TSO PU
```

To remove an access name from the Gateway database, use the CLEAR ACCESS NAME command with the ALL parameter. The following command removes the access name TSO:

```
SNANCP> CLEAR ACCESS NAME TSO ALL
```

8.5 Sample Access Name Task

The following example shows how to use SNANCP commands to manage access names for the sample network shown in Chapter 1. This example assumes you have invoked SNANCP and have specified the name of the Gateway you use. Note that you do not have to set up access names for all the access-routine products. Users can specify all access information when they are using the individual products (except for RJE, which requires access names). However, for this example, an access name is defined for each of the IBM applications available to Gateway users. For the sample network, you need to set up seven access names:

1. CICS — uses LUs 1 through 4 on PU SNA-0 for TE users to access CICS:

   ```
   SNANCP> SET ACCESS NAME CICS PU SNA-0 LU LIST 1-4 -
   _SNANCP> APPLICATION CICS NOTE "TE access to CICS"
   ```

2. TSO — uses LUs 5 through 8 on PU SNA-0 and LUs 1-25 on PU SNA-1 for TE users to access TSO:

   ```
   SNANCP> SET ACCESS NAME TSO PU SNA-0 LU LIST 5-8 -
   _SNANCP> SET ACCESS NAME TSO PU SNA-1 LU LIST 1-25 -
   _SNANCP> APPLICATION TSO NOTE "TE access to TSO"
   ```

3. IMS — Users invoke IMS to run user-written applications. IMS shares LUs 1–8 with the 3270 TE users. As well as specifying the application name, PU name, and LUs, this access name specifies IMSENT as the logon mode table to use for application-to-application programs:

   ```
   SNANCP> SET ACCESS NAME IMS PU SNA-0 LU LIST 1-8 APPLICATION -
   _SNANCP> IMS LOGON MODE IMSENT NOTE "LU0 applications"
   ```

4. RMT4 — uses LUs 9 through 12 for RJE users to access JES2. In addition to specifying the application name, PU name, and LUs, this access name specifies the user DATA for RJE as RMT4,LINE4,PWD4 and RJE1 as the logon mode table entry:

```
SNANCP> SET ACCESS NAME RMT4 PU SNA-0 LU LIST 9-12 APPLICATION -
_SNANCP> JES2 LOGON MODE RJE1 DATA "RMT4,LINE4,PWD4" -
_SNANCP> NOTE "RJE workstation RMT4"
```

5. DDXF — uses LUs 13 and 14 for DDXF users to access the CICS application DISOSS:

```
SNANCP> SET ACCESS NAME DDXF PU SNA-0 LU LIST 13,14 -
_SNANCP> APPLICATION CICS NOTE "DISOSS access"
```

6. SCSP — uses LU 18 for PrE users to use an LU Type 1 for DSPRINT:

```
SNANCP> SET ACCESS NAME SCSP PU SNA-0 LU LIST 18 -
_SNANCP> NOTE "PrE LU Type 1"
```

7. DSCP — uses LU 19 for PrE users to use an LU Type 3:

```
SNANCP> SET ACCESS NAME DSCP PU SNA-0 LU LIST 19 NOTE
"PrE, LU Type 3"
```

You can use the SHOW command to check the list of access names. In the following example, the user is a privileged user; therefore, the display shows DATA as well as the other parameters:

```
SNANCP> SHOW KNOWN ACCESS NAMES CHARACTERISTICS

Known Access name Characteristics as of 7-AUG-1988 11:54:40 from THYME

Access name = CICS

Note        = TE access to CICS
PU          = SNA-0
LU list     = 1-4
Application = CICS

Access name = DDXF

Note        = DISOSS access
PU          = SNA-0
LU list     = 13,14
Application = CICS

Access name = DSCP

Note        = PrE, LU Type 3
PU          = SNA-0
LU list     = 19

Access name = IMS

Note        = LU0 applications
PU          = SNA-0
LU list     = 1-8
Logon mode  = IMSENT
Application = IMS
```

```
Access name = RMT4

Note        = RJE workstation RMT4
PU          = SNA-0
LU list     = 9-12
Logon mode  = RJE1
Data        = RMT4,LINE4,PWD4
Application = JES2

Access name = SCSP

Note        = PrE, LU Type 1
PU          = SNA-0
LU list     = 18
appl        = CICS
Access name = TSO

Note        = TE access to TSO
PU          = SNA-0
LU list     = 5-8
PU          = SNA-1
LU list     = 1-25
Application = TSO
```

9 Server Management

This chapter describes how to manage servers on the Gateway. Managing servers involves defining server parameters, displaying information to monitor activities of the server, and clearing server parameters.

After installing Gateway software, you run a configuration command procedure that defines servers and server parameters (see *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*). These definitions are executed at load time. After the Gateway is loaded, you can use SNANCP commands to define additional server parameters and to change the values of parameters defined during installation or loading.

When you enter SNANCP commands interactively, any parameter definitions or changes you make with these commands stay in effect only while the Gateway is running. To cause new values to be used the next time the Gateway is reloaded, edit the following file that contains the default values for SNA components:

SYS$COMMON:[SNA$CSV]SNAGATEWAY_*gateway-node-name*_SNA.COM

9.1 Gateway Servers

There are three Gateway servers: DHCF, RJE, and SNA-ACCESS. (The SNA-ACCESS server is also known as the Gateway Access Server (GAS).) Gateway servers provide services required by the Gateway access routines. For example, the SNA-ACCESS server provides services for DECnet SNA DDXF, 3270 TE, API, APPC/LU6.2, 3270 DS, Message Router/S, Message Router/P, and PrE and DECnet SNA DTF access routines.

9.1.1 Server States

A server has two possible states: ON or OFF. A server must be in the ON state for an access routine to use the services provided by the server. For example, if the DHCF server is OFF, IBM HCF users cannot use it to access the DECnet /SNA VMS Distributed Host Command Facility (DHCF) access routine.

If your network does not need to use a particular server, set that server to OFF. Otherwise, that server will use Gateway resources unnecessarily.

9.1.2 Server Identification

A *server-id* value can be one of the following:

DHCF	Allows IBM 3270 terminals using the IBM HCF program to access DHCF on an OpenVMS host. For more information, see *DECnet/SNA Distributed Host Command Facility Management*.
RJE	Allows DECnet SNA Remote Job Entry for OpenVMS (RJE) to emulate an IBM 377*x* RJE workstation attached to an IBM JES, JES2, JES3, or VSE/Power host. For more information, see the *DECnet SNA Remote Job Entry for OpenVMS User's and Operator's Guide*.
SNA-ACCESS	Allows application programs, such as the DECnet SNA Printer Emulator (PrE) for OpenVMS, the DECnet SNA Data Transfer Facility (DTF), and the DECnet SNA 3270 Terminal Emulator (3270 TE) products, to access an SNA network.

9.1.3 Outbound Connection Initiation Through Servers

Initiating a session from within an SNA network to an object in a DECnet network is an example of an outbound connection. The Gateway allows you to make outbound connections (connections from a primary logical unit (PLU) within an SNA network to an object in a DECnet network) through LUs that are allocated to a server for this type of connection.

To enable outbound connection initiation, you must specify that certain LUs are allocated to a server for outbound connections. When the Gateway receives a BIND, it checks to see if the LU specified in the BIND is designated for outbound connection initiation. If the LU is defined for outbound connection initiation, the Gateway attempts to establish a session with the DECnet object associated with the LU.

Note

SNA-ACCESS is the only server that supports outbound connection initiation at this time. Refer to the documentation for the individual access routine products to see if the product you are using supports outbound connection initiation.

9.2 Defining Server Parameters

The SNANCP SET SERVER command defines parameters for a server.

Each server can impose specific restrictions on the maximum number of links that can be supported. If you specify a larger value than the supported maximum, the supported maximum is set.

9.3 Monitoring Servers

Before making changes to server parameters, you can check the current parameter values. The SNANCP SHOW SERVER command displays information about servers. By default, the information is sent to SYS$OUTPUT, which is usually your terminal. You can use the optional qualifier, TO *file-spec*, to send the information to a file.

If you enter the SHOW SERVER command without any parameters, a summary display is shown:

```
SNANCP> SHOW KNOWN SERVERS

Known Server Summary as of  1-AUG-1988 16:57:29 from THYME

  Server          State                     Note
  SNA-ACCESS      on                        Gateway Access Server
  DHCF            on                        Host Command Facility
  RJE             on                        Remote Job Entry
```

Since the displays for each server can vary because of server-specific parameters, Sections 9.3.1 to 9.3.3 discuss the SHOW SERVER command for individual servers. Section 9.3.4 describes the display for the COUNTER parameter.

For more information on the SNANCP SHOW SERVER command, see Chapter 13.

9.3.1 DHCF

The SHOW SERVER DHCF STATUS command displays information specific to DHCF if the server is ON. If the server is OFF, only the server name and its state are displayed. The following is a sample display for a server in the ON state:

```
SNANCP> SHOW SERVER DHCF STATUS

Server Status as of 1-AUG-1988 17:22:06 from THYME

Server                  = DHCF
State                   = ON
Logon text file state   = included

LU          Node     Terminal   Link       Session State

SNA-0.31                                   available
SNA-0.32    WOMBAT   SNA        43034      logged on
```

The following is an explanation of the fields in the display:

Node—Shows which DECnet node the IBM user is accessing through the Gateway.

Link—The number of the DECnet logical link.

Terminal—Shows whether the terminal is SNA or non-SNA.

State—An LU can be in any of the following states:

AVAILABLE	The LU is available for an HCF user to acquire.
NOT AVAILABLE	DHCF could not listen to the LU for one of several reasons: – The PU or LU might not have been defined with SNANCP SET commands. – The LU might have been acquired by another server. The reason for the failure is reported in the event log.
ACQUIRED	The LU has been acquired by an HCF user but there is not yet an associated DECnet logical link. At this point, the TERMINAL type is displayed. The terminal type is either SNA or non-SNA.
LOGGED ON	An HCF user has a DECnet logical link with an OpenVMS node. At this point, the display shows the DECnet node name and the DECnet logical link number.

The Logon text file state indicates whether a text file has been loaded into the logon screen buffer. Logon text file can be in any of the following states:

INCLUDED	Indicates that the logon screen includes the text.
NOT-INCLUDED	Indicates that the logon screen does not yet include the text.
ACCESS-ERROR	Indicates that DHCF was unable to read the file. The reason for an access error failure is reported in the event log, if event logging is enabled.

The following command displays DHCF characteristics:

```
SNANCP> SHOW SERVER DHCF CHARACTERISTICS

Server Characteristics as of 2-AUG-1988 13:46:18 from THYME
Server = DHCF

Logging enabled          = warning
Note                     = Host Command Facility
Maximum links            = 3
Logon text file          = WOMBAT::DISK$DHCF:[DHCF]LOGON.TXT
PU                       = SNA-0
LU list                  = 15-17
```

9.3.2 RJE

The SHOW SERVER RJE STATUS command displays information specific to RJE if the server is ON. If the server is OFF, only the server name and its state are displayed. The following is a sample display for a server in the ON state:

```
SNANCP> SHOW SERVER RJE STATUS

Server Status as of 1-AUG-1988 17:22:06 from THYME

Server                   = RJE
State                    = ON

LU             Node     Workstation ID         Link   WS state

SNA-1.1        WOMBAT   KEVIN34                43039  on
SNA-1.2        WOMBAT   KEVIN34                43039  on
SNA-1.3        WOMBAT   KEVIN34                43039  on
SNA-1.5        WOMBAT   KEVIN34                43039  on
SNA-1.6        WOMBAT   KEVIN34                43039  on
```

The following is an explanation of the fields in the display:

Node—Shows from which DECnet node the RJE user is accessing the Gateway.

Link—The number of the DECnet logical link.

Workstation ID—The workstation identification is provided by the RJE user.

WS state—The state of the workstation can be any of the following:

OFF	The workstation has not been set to the ON state by the RJE user.
ON	The workstation is active; data can be sent or received on the streams assigned to that workstation.
VERIFY	The RJE server and DECnet SNA RJE utility (MPX) are exchanging version numbers.
LOCKED	The RJE user has set the workstation OFF and the RJE server is terminating the sessions.
LOAD	The RJE server is loading the translation tables.

9.3.3 SNA-ACCESS

The SHOW SERVER SNA-ACCESS STATUS command displays information specific to SNA-ACCESS if the server is ON. If the server is OFF, only the server name and its state are displayed. The following is a sample display for a server in the ON state:

```
SNANCP> SHOW SERVER SNA-ACCESS STATUS

Server Status as of 28-JUL-1988 16:39:30 from LEDGE

Server                  = SNA-ACCESS
State                   = ON

LU       Node    Remote user  Terminal       Link   Session state

SNA-0.1  WOMBAT  PORTER       T1312A$PORT_3  43024  active
SNA-1.4  WALABY  PORTER       WTA2           43033  active
```

The following is an explanation of the fields in the display:

Node—The DECnet node from which the user is accessing the Gateway.

Remote user—The user currently accessing the LU. Older versions of the DECnet SNA access routines do not provide this information. If a user is accessing the Gateway with an access routine that does not support this information or if the accessing process has no controlling terminal, this field will not be shown.

Terminal—The terminal currently accessing the LU. Older versions of the DECnet SNA access routines do not provide this information. If a terminal is accessing the Gateway with an access routine that does not support this information, this field will not be shown.

Link—The number of the DECnet logical link.

Session state—A session can be in any of the following states:

PENDING	The logical link is being established.
CONNECTING	The Gateway is allocating the necessary resources.
NOT-BOUND	The LU–LU session is not active. However, the LU–SSCP session might be active.
ACTIVE	The LU–LU session is active and data can be sent and received.
ABORTING	The Gateway is terminating the LU–LU session and the logical link.
UNBINDING	The LU–LU session is being unbound.

Displaying Outbound Connection Information

Use the SHOW SERVER command with the LU component qualifier to display outbound connection information.

The following example shows the type of display the SUMMARY parameter creates:

```
SNANCP> SHOW SERVER SNA-ACCESS KNOWN LUS SUMMARY

Server known LU Summary as of  6-APR-1988 16:31:23 from THYME

  Server        LU         Node    Object Name  Object Number
  SNA-ACCESS    SNA-0.1    WOMBAT               129
  SNA-ACCESS    SNA-0.2    WALABY  MYOBJECT     0
  SNA-ACCESS    SNA-0.3    THYME                150
```

The following example shows the type of display you see if you use the CHARACTERISTICS parameter:

```
SNANCP> SHOW SERVER SNA-ACCESS LU SNA-0.1 CHARACTERISTICS

Server LU Characteristics as of  6-APR-1988 16:31:13 from THYME

Server = SNA-ACCESS

LU                        = SNA-0.1
Node                      = WOMBAT
Object number             = 129
User                      = JOHN
Password                  = password
Note                      = Gateway Access Server
```

9.3.4 Displaying Server Counters

Use the SNANCP SHOW SERVER command with the COUNTERS parameter to display counter information for the DHCF and SNA-ACCESS servers. There are no SNANCP counters for the RJE server. Counters for RJE are maintained by the host-based RJE multiplexer.

The following command displays counters for DHCF:

```
SNANCP> SHOW SERVER DHCF COUNTERS

Server Counters as of 28-AUG-1988 11:52:35 from THYME

Server = DHCF

      101339  Seconds since last zeroed
           0  Logical links active
           0  Total LUs in LU lists
           0  Maximum acquired LUs
           0  Current acquired LUs
           0  DECnet connect failures
           0  DECnet logical link failures
           0  Session protocol errors
           0  Internal software errors
           0  BINDs rejected
           0  Buffer allocation failures
```

9.3.5 Zeroing Server Counters

After viewing the counters, you can use the SNANCP ZERO SERVER command to set the SERVER counters to zero. For more information on the SNANCP CLEAR SERVER command, see Chapter 13.

The following command zeros all the counters for the SNA-ACCESS server:

```
SNANCP> ZERO SERVER SNA-ACCESS
```

If event logging is enabled for informational messages, a message is logged to the DECnet host when you set counters to zero.

9.4 Clearing Server Parameters

After viewing information about server parameters, you can change the values of one or more parameters. Use the CLEAR or SET commands:

- CLEAR—Resets parameters to their default value.
- SET—Provides new values for parameters.

For DHCF, set the server state to OFF before you clear the MAXIMUM LINKS, and PU LU LIST parameters. For all servers, set the state to OFF before using the CLEAR ALL command to reset all parameters to their default values. The CLEAR ALL command, (unlike CLEAR ALL for other Gateway components) does not remove a server definition; it merely sets all parameters to their default values.

The following command resets the LUs for DHCF:

```
SNANCP> SET SERVER DHCF STATE OFF
SNANCP> CLEAR SERVER DHCF PU SNA-0 LU LIST
```

For more information on the SNANCP CLEAR SERVER command, see Chapter 13.

Clearing Outbound Connection Parameters

The following command clears all outbound connection parameters for LU SNA-0.1:

```
SNANCP> CLEAR SERVER SNA-ACCESS LU SNA-0.1 ALL
```

9.5 Sample Server Tasks

Task 1

The following is an example of how to use SNANCP commands to define and manage servers for the sample network shown in Chapter 1. This example assumes that you have invoked SNANCP and have specified the Gateway you will use.

```
SNANCP> SET SERVER SNA-ACCESS STATE OFF
SNANCP> SET SERVER SNA-ACCESS STATE ON MAX LINKS 13 -
_SNANCP> LOGGING WARNING
SNANCP>
SNANCP> SET SERVER RJE STATE OFF
SNANCP> SET SERVER RJE STATE ON MAX LINKS 1 -
_SNANCP> LOGGING WARNING
SNANCP>
SNANCP> SET SERVER DHCF STATE OFF
SNANCP> SET SERVER DHCF STATE ON MAX LINKS 3 PU SNA-0 -
_SNANCP> LU LIST 15-17 TEXT FILE SNAGWL::DISK$DHCF:[DHCF]LOGON.TXT -
_SNANCP> NOTE "LUS 15-17" LOGGING WARNING
```

Next, a user on SNAGWB assigns a reader stream and a printer stream to an RJE workstation. You can use SNANCP to look at the status of the workstation:

```
SNANCP> SHOW SERVER RJE STATUS

Server Status as of 3-AUG-1988 14:42:08 from SNAGWY

Server                  = RJE
State                   = on

LU             Node     Workstation ID             Link   WS state

SNA-0.9        SNAGWB   RMT4                       4      on
SNA-0.10       SNAGWB   RMT4                       4      on
SNA-0.11       SNAGWB   RMT4                       4      on
```

You can use the SNANCP SPAWN command to create a subprocess, and then use NCP commands in the subprocess to look at the DECnet link:

```
SNANCP> SPAWN RUN SYS$SYSTEM:NCP
NCP>SET EXECUTOR NODE SNAGWY
NCP>SHOW LINK 4

Link Volatile Summary as of 3-AUG-1988 14:43:40

Link      Node           PID     Process     Remote link  Remote user
4         1.3 (SNAGWB)   0801ADB0  SNA$RJE         9268      SYSTEM

NCP>EXIT
SNANCP>
```

The NCP SHOW LINK 4 command displays one workstation connected to the DECnet SNA RJE multiplexer program running on SNAGWB.

Task 2

This task presents a situation in which outbound connections are used to save system resources. A user has written a program called DAILY_UPDATE. This program uses APPC/LU6.2 and its purpose is to receive a connection from a program called CENTRAL_DB on the IBM system. Once a connection is established, the program reads records from a database on the local DECnet system and transmits this data to the cooperating task on the IBM system. This transaction takes about five minutes and occurs once a day. It is a waste of resources to have the DAILY_UPDATE program connected to the Gateway all day waiting for five minutes of activity.

To avoid wasting system resources, allocate an LU (associated with the SNA-ACCESS server) for outbound connections. To allocate an LU, you must provide the following information:

- Which LU is used for communication between the IBM program and the program on the DECnet side?
- On which DECnet node will the OpenVMS program run?
- What type of access control is needed?

- What is the number of the DECnet object?

The following example lists the steps involved in establishing outbound session allocation for LU SNA-1.64, for the program DAILY_UPDATE running on node HAVEN. No access control is needed, and the object number is 150:

1. Use an NCP command to define DAILY_UPDATE as an object on the DECnet node on which it is running. See the *OpenVMS Network Control Program Manual* if you need information on defining DECnet objects.
2. Use the following SNANCP command to associate LU SNA-1.64 with DAILY_UPDATE:

   ```
   SNANCP> SET SERVER SNA-ACCESS LU SNA-1.64 NODE HAVEN OBJECT 150
   ```

3. Use the SNANCP SHOW command to verify the association:

   ```
   SNANCP> SHOW SERVER SNA-ACCESS LU SNA-1.64 CHARACTERISTICS
   ```

10 Event Logging

Event logging records significant events that occur while a Gateway system is running. The Event Logger monitors the following types of events:

- Circuit or line counter values before the counters are zeroed
- Changes in circuit, line, and node states (for example, a circuit failure)
- Passive loopback (when executor is looping back test messages)
- An attempt to access an LU by an unauthorized user

There are two kinds of event logging for a Gateway system: DECnet event logging and Gateway event logging. A Gateway load host node or access node that has Gateway management software installed can record event-logging information for both OpenVMS nodes and the Gateway. DECnet event logging for OpenVMS nodes is described in the *OpenVMS Networking Manual*; the information is not repeated in this chapter. Gateway event logging works in a similar way.

A node on which events are logged is called a *sink node*. A device or process that records logging events on a sink node is called a *logging component*. There are three different logging components you can use: logging file, logging monitor, and logging console. You use SNANCP commands to specify the sink node(s) and logging component(s) to which you want events logged. You can also use SNANCP commands to specify the type of events you want logged by selecting a minimum level of severity. The sections that follow describe Gateway event logging and the SNANCP commands that set logging parameters.

10.1 Starting the Event Logger

Any node on which you log events must be a Gateway load host or Gateway access node that is running Gateway management software. The node must also be running the SNA Event Logging program (SNAEVL). If you did not start the event logger after installing the Gateway management software, start it now using the following command:

```
$@SYS$STARTUP:SNAGM$EVL_STARTUP
```

If you include the preceding command in the file SNAGM$STARTUP (located in SYS$STARTUP), the SNA event logger will start automatically each time you boot the system.

10.2 Gateway Logging Commands

There are two types of SNANCP logging commands for the Gateway: those that specify a logging component such as a logging console and those that specify a Gateway component, such as a line.

The functions of the two types of commands are as follows:

- SET LOGGING *logging-component parameter-list*

  Lets you:

  - Specify what logging components to use
  - Specify sink nodes
  - Turn logging on or off

  For example, the following command turns on logging for the named logging file. (The default sink node, the load host, is implied in the example).

  ```
  SNANCP> SET LOGGING FILE NAME SYS$MANAGER:SNAEVL.DAT STATE ON
  ```

- SET *gateway-component* LOGGING [ENABLED] *parameter-list*

  Lets you specify the severity of events that are logged for a particular Gateway component, such as a line. For example, the following command specifies that all events severity level of ERROR or greater are logged for the SNA-ACCESS server:

  ```
  SNANCP> SET SERVER SNA-ACCESS LOGGING ERROR
  ```

The sections that follow discuss the two types of SNANCP logging commands in more detail.

10.3 Where to Log Event Messages

By default, the Gateway sends event messages to the DECnet logging monitor OPCOM (OPerator's COMmunications facility) on the load host. Directing the event log here is useful if an operator needs to know what is occuring on the network, as it occurs.

Event messages sent to OPCOM are displayed at all operator terminals for operator class NETWORK and are stored in the operator log file. To enable a terminal as a network operator, enter the following command:

```
$ REPLY/ENABLE=NETWORK
```

You will then receive all network event messages at that terminal. (All Gateway event messages are network messages.) You will receive other DECnet network messages as well as the Gateway event messages. Gateway event messages are identified by the text "DECnet/SNA". Appendix D explains the event codes.

If you want to change the default values for Gateway logging, you can specify which Gateway access nodes will log events and whether the messages are recorded in a binary file, a logging console, or a logging monitor. The following command specifies that event messages are logged on NODEB at the default monitor OPCOM (in addition to being logged to the load host):

```
SNANCP> SET LOGGING MONITOR SINK NODE NODEB STATE ON
```

If you want to specify that event messages are logged at NODEC also, use a second SET LOGGING command that specifies NODEC as the SINK NODE. Messages are then logged at both NODEB and NODEC. The effects of the SET LOGGING command are cumulative.

If the LOGGING MONITOR is enabled, then events are always sent to OPCOM on the sink node. If you also specify a NAME for the monitor component, events will be written to the named monitor process in addition to being sent to OPCOM.

To log event messages to a logging console, specify LOGGING CONSOLE as the logging component in your SET LOGGING command. A logging console is a display of events in text form at a dedicated terminal or in a specified file. If you specify a terminal, it must be dedicated exclusively to the event logger. The next example sets console logging on the load host and sends the output to a file.

```
SNANCP> SET LOGGING CONSOLE STATE ON NAME SYS$MANAGER:SNAGWY.LOG
```

Additionally, you can have the event messages sent to a binary file by using LOGGING FILE. The information in binary files is for the use of Digital software specialists only. They use the binary file if you have a problem with the Gateway that you are unable to solve. For normal use of logging events, specify LOGGING MONITOR or LOGGING CONSOLE.

In addition to specifying which node and which logging component will display messages, you can control the operational state of logging for the Gateway. The logging of event messages begins as soon as you set the logging state to ON. The following command sets event logging for all sink nodes to the ON state:

```
SNANCP> SET KNOWN LOGGING STATE ON KNOWN SINKS
```

10.4 Specifying Severity for Event Messages

The SNANCP SET *component-name* LOGGING [ENABLED] command lets you specify the level of severity and the gateway component for logged events. The following command logs all informational, warning, error, and fatal event messages relating to lines:

```
SNANCP> SET KNOWN LINES LOGGING INFORMATIONAL
```

The severity level you specify in the command sets the minimum level of message severity. For example, when you specify INFORMATIONAL severity, messages of all levels of severity are logged. If you specify WARNING then warning, error, and fatal event messages will be logged but informational messages will not. The levels of severity follow; they are listed from least severe to most severe. Fatal messages are always logged.

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

10.5 Displaying Event Logging Information

You can look at the current values for event logging by using the SHOW LOGGING command.

If you do not enter a parameter to specify a type of display, a summary display is shown by default:

```
SNANCP> SHOW KNOWN LOGGING

Known Logging Summary as of 2-AUG-1988 17:44:24 from THYME

 Logging sink type = monitor

    Sink Node       State                 Name

    1.10 (NODEG)    ON
```

In the display, the State field indicates whether you have set logging ON or OFF. The Name field contains a file specification if you are logging events in a file.

If you enter the CHARACTERISTICS parameter, you will see information about the parameter values of event logging.

To see the severity of event messages for a particular component, use the SHOW *component-name* CHARACTERISTICS command.

The following command shows the logging severity levels for line SYN-0 as part of its display:

```
SNANCP> SHOW LINE SYN-0 CHARACTERISTICS
```

For more information on the SNANCP SHOW LOGGING command, see Chapter 13.

10.6 Clearing Event Logging

There are two forms of the CLEAR command for logging:

- CLEAR LOGGING affects all event logging.
- CLEAR *component-name* LOGGING affects the severity of messages logged for a particular component.

CLEAR LOGGING Command

If you do not specify SINK NODE or KNOWN SINKS, the default sink (the load host) is used. SINK NODE *node-id* causes specified logging to be cleared only on the named sink node. KNOWN SINKS causes specified logging to be cleared on all known sinks.

You must set event logging to the OFF state before you can clear it. If you specify the ALL parameter, it removes the specified logging components.

If you use the NAME parameter, the CLEAR LOGGING command removes the name the console, file, or monitor process (other than OPCOM) to which messages are logged.

The following command clears the name of the monitor that logs the event messages; logging messages are then sent only to OPCOM.

```
SNANCP> CLEAR LOGGING MONITOR SINK NODE NODEG NAME
```

CLEAR component-name LOGGING Command

To clear event logging for a particular component, use the CLEAR *component-name* LOGGING command.

The following command clears all the event logging (except for fatal messages) for the line SYN-0:

```
SNANCP> CLEAR LINE SYN-0 LOGGING
```

After you have used the previous CLEAR command to disable event logging, the SHOW LINE CHARACTERISTICS display will show fatal event logging as the only type enabled:

```
SNANCP> SHOW LINE SYN-0 CHARACTERISTICS

Line Characteristics as of 21-JUN-1988 14:11:34 from THYME

Line = SYN-0

Counter timer            = 0
Receive buffers          = 64
Duplex                   = full
Protocol                 = SDLC Point
Clock                    = external
Buffer size              = 1420
Signalling convention    = normal
Modem type               = normal
Logging enabled          = fatal
```

11

Using NETTRACE

There are two types of traces you can use to isolate a Gateway problem:

- **IBM traces**. You can use data from IBM traces to show protocol sequences on the line level, on the PU level, and on the LU level. You can also request a VTAM internal trace and use the results to determine which control blocks VTAM used when establishing a session. For more information on IBM traces, refer to *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)* or *DECnet SNA Gateway for Synchronous Transport Problem Solving*.
- **Gateway management traces**. Gateway management software provides two protocol trace utilities, SNATRACE and NETTRACE. These utilities show the SNA protocol exchanges between the Gateway and an IBM system. You can use SNATRACE or NETTRACE to isolate problems associated with the Gateway and its DECnet SNA access routines.

SNATRACE is described in Chapter 12. NETTRACE includes the following features:

- Multiple session tracing, enabling you to trace more than one session at a time
- Detached process tracing, which allows you to use your terminal for other tasks while trace data is being collected in a file
- Numerous analysis options, that enable you, for example, to specify whether you want the data to appear in hexadecimal, EBCDIC, or ASCII format
- Output formatting in columns, enabling you to quickly find information
- Decoding of CHAN and SDLC fields, revealing data link information such as polling status and frame type

When you start a trace you specify tracepoints from which you want to collect data. The three tracepoints are CHAN, SDLC, and PU.

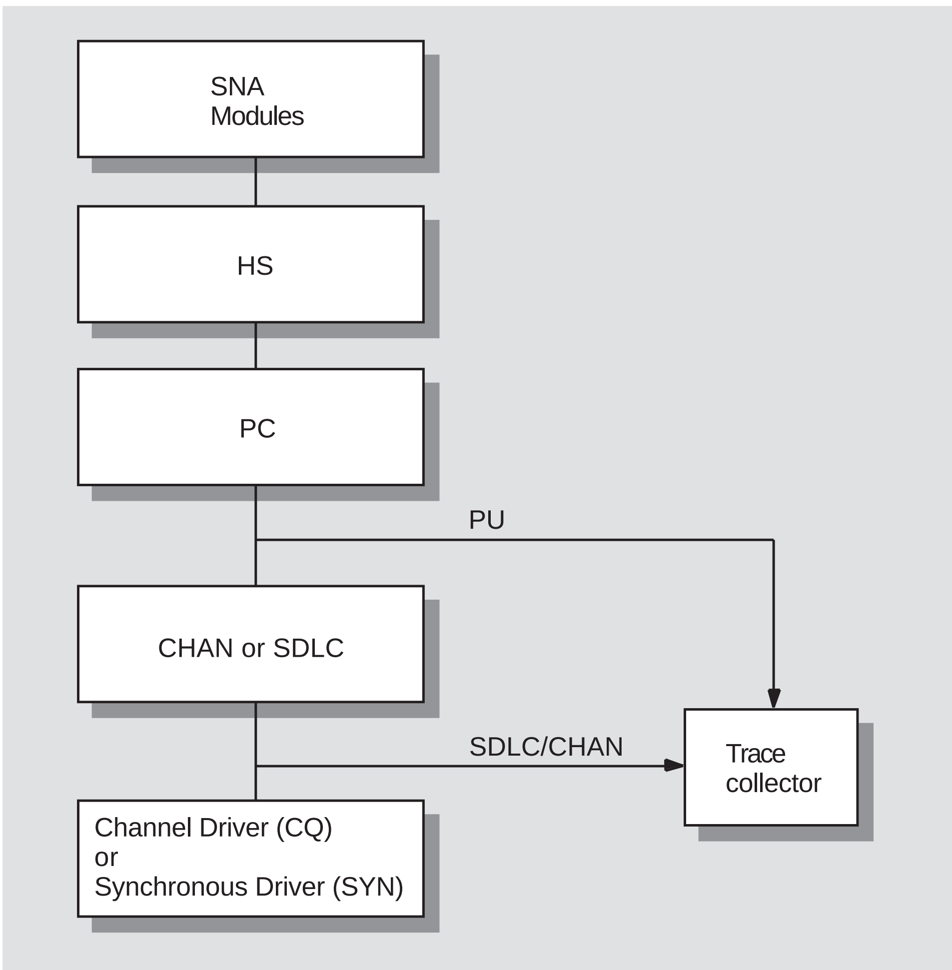
CHAN and SDLC are data link level (or circuit level in DECnet terms) tracepoints. The CHAN tracepoint traces a channel-attached circuit on a Gateway-CT. The SDLC tracepoint traces a synchronous-line circuit on a Gateway-ST. If you suspect a problem in the data link communications between the Gateway and the IBM side, start a trace specifying the SDLC or CHAN tracepoints.

The PU tracepoint captures data that passes between the path control (PC) layers in the gateway and the IBM side. If you suspect a problem in communications at the path control layer, start a trace specifying PU as the tracepoint.

Session level tracing is also supported by NETTRACE. A session is not a separate tracepoint. NETTRACE filters the PU tracepoint data to show the session flows if you specify the session option in the command to start the trace.

Figure 11–1 shows a trace collector receiving data at all valid tracepoints. A trace collector is the node and process that has issued the command to start a trace. This is where the trace data are collected for you to analyze.

Figure 11–1 Tracepoints



LKG−4149−93R

11.1 Privileges Required for NETTRACE

To use the SHOW command in NETTRACE, you need the TMPMBX and NETMBX privileges. In addition, if you want to perform either the START or STOP commands, you need the OPER privilege.

To use the ANALYZE command, you need no additional privileges, but you need access to the file that contains the trace data.

When you specify a gateway node in the SHOW and START commands, you must also specify the privileged username and password associated with that gateway.

11.2 Starting NETTRACE

To run NETTRACE, issue the DCL command (for DECnet/VAX):

```
$ TRACE
```

and for DECnet/OSI:

```
$ TRACE/SNA
```

The system prompts you as follows:

```
NETTRACE>
```

You can now enter any NETTRACE command.

You can also enter the ANALYZE, HELP, SHOW, START, and STOP commands along with the NETTRACE command at the DCL prompt. For example, for DECnet/VAX:

$ TRACE START *gateway-name*"SYSTEM *password*"::PU

and for DECnet/OSI:

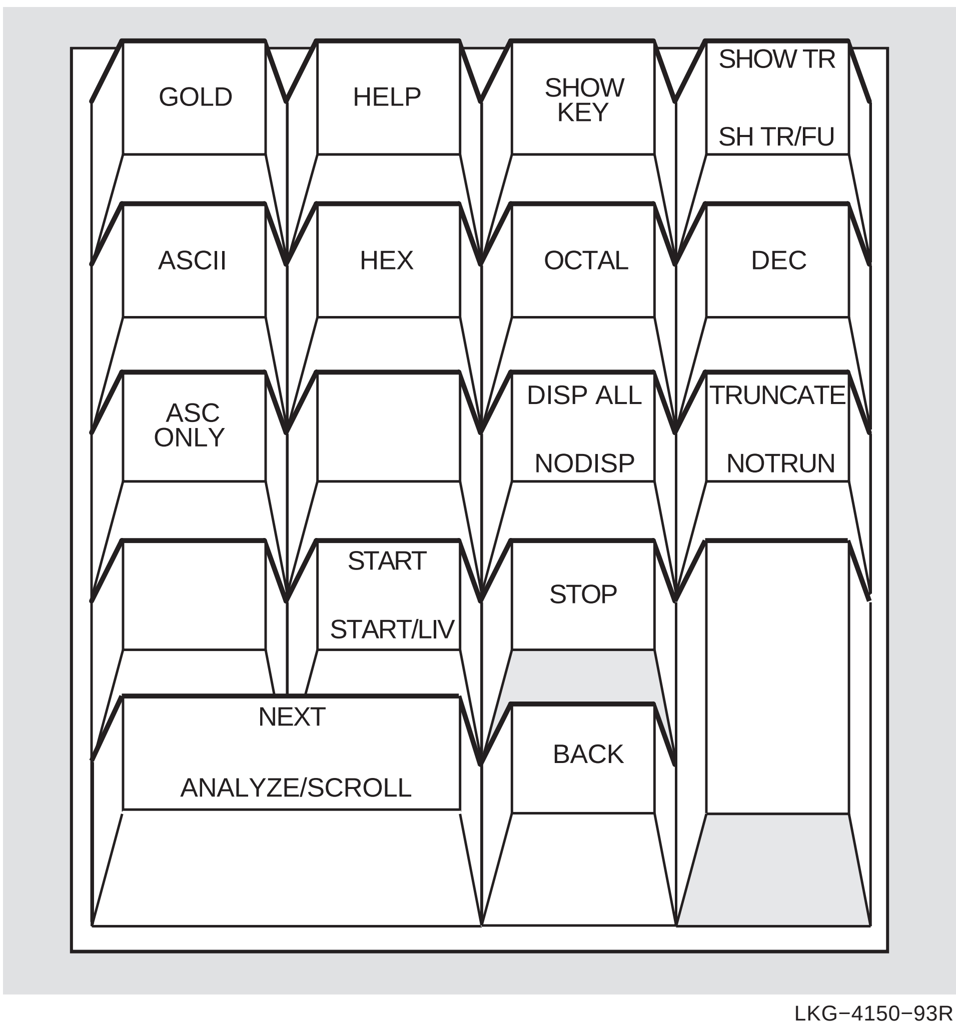
$ TRACE/SNA *gateway-name*"SYSTEM *password*"::PU

To enter a NETTRACE command, either enter the command in full (adding optional qualifiers where necessary), or use the default NETTRACE keypad. The default NETTRACE keypad is set up when NETTRACE is run. Use Help to obtain help on the default keypad definitions. You can add new key definitions by using the DEFINE/KEY command (and delete existing definitions by using the DELETE/KEY command). DEFINE/KEY allows you to associate a string and set of attributes with a key on the keyboard.

Note

To display a list of the currently defined keys, use the SHOW KEY/ALL command. To see which commands correspond to keys on the default keypad, see Figure 11–2.

Figure 11–2 NETTRACE Keypad



In addition to the keys shown on the keypad, the file from which the keypad is set up provides definitions for the following keys:

- Gold/Next Screen as ANALYZE/SCROLL
- Next Screen as a NEXT screen command
- Prev Screen as a BACK screen command

- Help as a HELP command
- Ctrl/L as a CLEAR screen command
- Ctrl/W as a REFRESH screen command

There are two methods of collecting data using the NETTRACE utility:

1. You can collect and analyze the data while it is traced. This method is known as LIVE tracing (see Section 11.2.1).
2. You can record binary data in a file and analyze it later. For this method, you need to run NETTRACE as a detached process (see Section 11.2.2).

Note

When you are using the NETTRACE utility, data often arrives faster than NETTRACE can capture, analyze, or display it on the screen. NETTRACE can lose trace records.

11.2.1 Performing LIVE Tracing

You can initiate the collection and display of data at your terminal by issuing a TRACE command followed by a START/LIVE command. For example:

```
$ TRACE
NETTRACE> START/LIVE gateway-node-name"username password"::CHAN
```

will start tracing on all defined CHAN circuits on the gateway you specify.

After you issue the command, you will see the following message on your screen:

```
%NETTRACE-I-CHANSTRACED, n channels are currently being traced
```

where n is the number of lines or channels currently being traced. If new lines or channels are created while tracing is running, these will also be traced. You can use the SHOW command to display the names of the channels being traced (see Section 11.3.11).

You can now issue an ANALYZE command to alter the format of the data output. For example:

```
NETTRACE> ANALYZE/DATA=ASCII
```

changes the data output to ASCII format.

When you want to stop tracing, exit from NETTRACE by typing Ctrl/Z. Tracing stops, tracepoints are deleted, and the DCL prompt returns. If you have set up more than one tracepoint, you cannot turn them off individually. Also, a tracepoint cannot exist while no trace collector is working.

You can issue SHOW commands to display the data being traced. For example:

```
NETTRACE> SHOW TRACE_COLLECTOR BORIS"TONY SECRET"::*/FULL
```

The following information is displayed:

```
Tracing on node BORIS:: on date-time

Collector       Collecting to    Tracepoints

BORIS::TONY     Interactive      CHAN
                                 CHAN-0
```

"Collector" is the node and process name that is collecting the trace data. The process name is the process where the START/LIVE command was issued. "Collecting to" shows whether tracing is being performed as an interactive process or whether data is being written to a file.

"Tracepoints" shows the tracepoints and the names of the lines or channels being traced. In this case, tracing is performed on circuit CHAN-0.

11.2.2 Running NETTRACE as a Detached Process

If you do not specify the /LIVE qualifier with the START command, a detached process collects the data and writes it to a file. This local trace collector has a name based on your username taking the format *username*$NETTRACE[_*n*], where *n* is an integer. (However, if *username*$NETTRACE[_*n*] has more than 15 characters, the string NETTRACE is truncated to reduce the name to 15 characters.)

You initiate trace collection by issuing a TRACE START command. For example, for DECnet/VAX:

```
$ TRACE START gateway-node-name"username password"::CHAN
```

and for DECnet/OSI:

```
$ TRACE/SNA gateway-node-name"username password"::CHAN
```

The following messages are displayed:

```
%NETTRACE-I-DETSTART, Detached trace collector JOE$NETTRACE has been
                      started
%NETTRACE-I-CREATEFILE, File DUA0:[JOE]TRACE.DAT;1 has been created
%NETTRACE-I-CHANSTRACED, 1 channel is currently being traced
```

These messages mean that tracing has started at the CHAN tracepoint. Trace data is written to TRACE.DAT in your current default directory (unless you have specified the /OUTPUT qualifier).

Shared file access allows you to analyze the file while data is collected.

As with live tracing, you can issue SHOW commands to display what is being traced. For example:

```
NETTRACE> SHOW TRACE_COLLECTOR gateway"username  password"::*/FULL
```

displays the following information:

```
Tracing on node node-name on date-time

Collector         Collecting to          Tracepoints

SMITH$NETTRACE    DUA0:[SMITH]TRACE.DAT  CHAN
                                         CHAN-0
```

"Collector" is the node and process name that is collecting the trace data.

"Collecting to" shows the name of the file to which data is being written. "Tracepoints" shows the tracepoints and the names of the lines or channels being traced. In this case, tracing is performed at the CHAN tracepoint on circuit CHAN-0.

If you exit NETTRACE at this point, tracing continues. To stop tracing, issue the STOP command. For example:

```
NETTRACE> STOP
```

returns the message:

```
%NETTRACE-I-LTCEXIT, Trace collector SMITH$NETTRACE has exited
```

Stopping NETTRACE deletes the tracepoints. You do not specify a node name in the STOP command; the process you are stopping must be on your local node.

Analyze the file by issuing an ANALYZE command. For example:

```
$ TRACE ANALYZE/OUTPUT=TRACE/DISPLAY=(TIME,EVENT,NAME)
```

or for DECnet/OSI:

```
$ TRACE/SNA ANALYZE/OUTPUT=TRACE/DISPLAY=(TIME,EVENT,NAME)
```

displays the following messages:

```
%NETTRACE-I-PROCFILE, Processing file DUA0:[SMITH]TRACE.DAT;1
%NETTRACE-I-CREATEFILE, File DUA0:[SMITH]TRACE.LIS;1 has been created
%NETTRACE-I-PROC_TRPT, Processing CHAN tracepoint
%NETTRACE-I-EOFILE, End of input reached
```

This command analyzes the file TRACE.DAT and writes the formatted output to TRACE.LIS.

Note

By default, the ANALYZE command starts at the lowest version number of the file that exists. If you want to analyze a file with a higher version number, you must specify the version number with the ANALYZE command.

The ANALYZE command takes the file as input and formats it according to the command qualifiers (see Section 11.3.1).

11.3 NETTRACE Commands

Table 11–1 contains a summary of available NETTRACE commands. These commands are discussed in the following sections. For ease of reference, the commands have been arranged in alphabetical order.

Table 11–1 NETTRACE Commands

Command	Description
ANALYZE	Analyzes the data in the file, or enables you to modify the analysis options.
ATTACH	Enables you to switch from one process to another.
BACK	Displays the last screen of analyzed data.
CLEAR	Clears the screen of analyzed data.
DEFINE/KEY	Associates a string and set of attributes with a key on the keyboard.
DELETE/KEY	Deletes the key definitions you have defined.
EXIT	Returns you to the DCL prompt.
HELP	Displays the online help about NETTRACE.

(continued on next page)

Table 11–1 (Cont.) NETTRACE Commands

Command	Description
NEXT	Displays the next screen of analyzed data.
REFRESH	Recreates the screen display.
SHOW	Displays the trace collector and tracepoints or displays the key definitions.
SPAWN	Creates a subprocess of the current process.
START	Starts data collection.
STOP	Stops data collection.

11.3.1 ANALYZE

The ANALYZE command reads the trace data in a specified trace data file and formats it to help you interpret the data. (D) indicates a default qualifier.

ANALYZE [*filename*] [*qualifiers*]

where

filename	Is the name of the file you wish to analyze. By default, the ANALYZE command starts at the lowest version of the file that exists and analyzes each version of the file in turn.

If you do not specify a file name, TRACE.DAT is the default.

While a file is being analyzed, you can issue the ANALYZE command with qualifiers but no filename. The ANALYZE qualifiers are changed; a new file is not opened.

ANALYZE qualifiers

The following qualifiers are available for use with the ANALYZE command.

/BEFORE[*=time*]

Allows you to see data traced before a specified time. Specify the time in standard OpenVMS format.

/DATA=(*data-type,...*)
/NODATA

Allows you to specify the format of the output, which can be one or more of the following: HEXADECIMAL, OCTAL, ASCII, EBCDIC, or DECIMAL. In ASCII output, control characters are represented by " ^character". For example: Ctrl/M is represented by "^M". The default is HEXADECIMAL.

/DISPLAY=(*display-option,...*)
/NODISPLAY

Allows you to specify the fields to appear in the analyzed output (see Table 11–2). The default display options are TIME, EVENT and SIZE.

Table 11–2 DISPLAY Qualifier Options

Option	Description
[NO]ALL	Specify ALL if you require all the display options listed below. Use NOALL to cancel all options.
[NO]EVENT	Shows whether this is a received or transmitted data buffer or some other event.
[NO]FLAG	A flag word in the trace record. Not normally used.
[NO]FUNCTION_CODE	The function code of the operation being traced, normally, "Data".
[NO]NAME	Name of the traced line or channel (particularly useful when more than one line or channel is being traced simultaneously).
[NO]SIZE	Shows the total size of the trace record in bytes. This figure includes all protocol headers but not CRCs or flags.
[NO]STATUS	The status field. Sometimes used to show whether the I/O operation completed successfully.
[NO]TIME	Displays the time at which the record was created (that is, the time at which data first passed through the protocol level being traced).

/EVENT=(*event-type,...*)

Where *event-type* is one or more of the following: TRANSMIT, RECEIVE, or ALL. This qualifier allows you to select the events to be included in the output. The default is ALL.

/NAME=*channel-name*

Where *channel-name* is the name of the trace records that are to be analyzed. If you specify only part of the channel name, then all records that start with this string will be selected for analysis. Use the /DISPLAY=NAME option to display the channel name in the trace record. /NAME=* will display all the records.

/OUTPUT=*output-filename*

Is the name of the output listing file. If you do not specify /OUTPUT, output is written to your terminal.

/PAGE=*number*

Allows you to specify the number of lines generated per page.

/PROTOCOL=*(protocol-identifier,...)*
/NOPROTOCOL

Where *protocol-identifier* is one or more of the tracepoints. Use /NOPROTOCOL if you want the protocol headers to appear as part of the displayed data (that is, without being formatted). The default protocols depend on which tracepoints are being analyzed. /PROTOCOL (not normally used) allows you to specify which protocol is displayed. (See Table 11–3 for tracepoints and the default protocol identifiers.)

Table 11–3 Default Protocol Identifiers

Tracepoint Name	Default Protocol
CHAN	CHAN
SDLC	SDLC
PU	PU

/REVERSE
(D) /NOREVERSE

If your terminal supports reverse video (and is set to DEC_CRT), specifying /REVERSE displays received data in reverse video and displays the titles in bold.

/SAVE_BUFFER_SIZE=*n*

Where *n* is the number of screens of saved lines of data. These lines can be examined by using the NEXT and BACK commands. The default is 30.

/SCROLL
(D) /NOSCROLL

Determines whether data is typed a page at a time or continuously. If you specify /NOSCROLL, issue the NEXT or BACK command to display the next screen. /NOSCROLL is not valid if output is to a file. ANALYZE/SCROLL will display data continuously following a NEXT or BACK command.

/SELECT=*(protocol-identifier,..)*
/NOSELECT

Where *protocol-identifier* is one or more of the tracepoints. This qualifier allows you to specify which protocol headers appear in the output. Any protocol headers that you SELECT are removed from the analyzed data. The default is to display all the protocol headers (see Table 11–3).

(D) /SESSION=(*session-range,...*)
/NOSESSION

Where *session-range* is a list of SNA session numbers and ranges. Use /NOSESSION to analyze trace records for all sessions. /SESSION allows you to specify which sessions to analyze. A session range can be a single value, range of values, or every session up to the maximum. For example, /SESSION=1,10-* specifies session 1 and all sessions greater than or equal to 10. This qualifier is valid for PU trace records only.

/SINCE=[*time*]

Allows you to see data traced since a specified time. Specify the time in standard OpenVMS format.

/TRACE_LEVEL=*(protocol-identifier,...)*

Where *protocol-identifier* is one or more of the tracepoints. If you have specified more than one tracepoint to enable simultaneous tracing of more than one protocol level, the analyzer will format trace records only for one of the protocols. By default, the analyzer formats the trace records of the first protocol level found in the file. Use /TRACE_ LEVEL to specify a protocol level. The trace records of the protocol you specify will then be formatted. (See Table 11–3 for tracepoints and the default protocol identifiers.)

/TRUNCATE
(D) /NOTRUNCATE

Determines whether data that will not fit on one line is truncated or printed on the next line.

/WIDTH=*number*

Allows you to specify the number of columns generated per page. If the data is written to a file, the default is 132. If the data is displayed on your terminal, the default is the width of your terminal screen.

11.3.2 ATTACH

The ATTACH command enables you to switch control from your current process to another. You must specify one: processname, /IDENTIFICATION, or /PARENT. The format of the command is as follows:

Syntax:

ATTACH [*process-name*]

where

process-name is the name of the process to which you want to attach.

You cannot connect to a process if:

- The process is your current process.
- The process is not part of your current session.
- The process does not exist.

If you specify a process-name, you cannot use the /IDENTIFICATION or /PARENT qualifier.

ATTACH qualifiers

The qualifiers you can use with the ATTACH command are as follows:

/IDENTIFICATION

Specifies the process identification (PID) of the process to which you wish to attach. If you specify /IDENTIFICATION, you cannot specify /PARENT or process name.

/PARENT

This qualifier is valid only if used from a subprocess. It allows you to attach to the parent process. If you specify /PARENT, you cannot specify /IDENTIFICATION or process name.

11.3.3 BACK

The BACK command allows you to move back through analyzed NETTRACE output one screen at time. See also the NEXT command (Section 11.3.9).

Syntax:

BACK

If you are running NETTRACE as a detached process, the output is directed to a file for later analysis. When the contents of the file are analyzed, the BACK command can be used to turn off the (default) scroll mode and display the data one screen at a time. Each time you enter the command, the previous screen is displayed.

During live tracing, initial use of the BACK command suspends NETTRACE activity and turns off scrolling. Thereafter, the command functions in the normal way, allowing you to move back through previous screens (one screen at a time).

To resume live tracing, first issue the NEXT command, and then type

```
ANALYZE/SCROLL
```

In this way, all saved records are scrolled through (on screen) before tracing begins again.

11.3.4 CLEAR

The CLEAR command clears the screen of analyzed data.

Syntax:

CLEAR

This command has the same effect as Ctrl/L.

11.3.5 DEFINE/KEY

The DEFINE/KEY command allows you to define a key on the keyboard with a command.

Syntax:

DEFINE/KEY *key-name equivalence-string*

where

key-name	Is the key you wish to define.
equivalence-string	Specifies the string to be processed when you use the defined key. If the string contains any spaces, the string must be enclosed in quotation marks.

DEFINE/KEY qualifiers

(D) indicates a default qualifier.

(D) /ECHO
/NOECHO

Specifies whether the command line is echoed after you enter the defined key. You cannot specify both /NOECHO and /NOTERMINATE for a key definition.

/IF_STATE= *state-list*
/NOIF_STATE= *state-list*

Specifies a list of states. Set any state to enable the specified key definition. For example, you can use the GOLD key to define a key.

/LOCK_STATE
(D) /NOLOCK_STATE

Retains the state set by the /SET_STATE qualifier until the /SET_STATE qualifier is used again to alter the state.

/SET_STATE=*state*
/NOSET_STATE=*state*

Associates a state with the key being defined. The state name can be any alphanumeric string. You cannot define a key by specifying both /SET_STATE and /TERMINATE.

/TERMINATE

(D) /NOTERMINATE

Determines whether the specified command string executes when you enter the key. When you use /NOTERMINATE, you must enter Return to execute the command. You cannot define a key by specifying both /SET_STATE and /TERMINATE.

For further information on key definitions and states, refer to the *OpenVMS DCL Dictionary*.

11.3.6 DELETE/KEY

The DELETE/KEY command allows you to delete a key definition.

Syntax:

DELETE/KEY *key-name*

where

key-name	Is the key you wish to delete.

DELETE/KEY qualifier

/STATE=*state-name[,...]*

/NOSTATE=*state-name[,...]*

Specifies the name(s) of state(s) for which the specified key definition(s) are to be deleted. If you specify only one state name, you can omit the parentheses. A state name can be any appropriate alphanumeric string. If you omit the /STATE qualifier or use /NOSTATE, key definitions in the current state are deleted.

11.3.7 EXIT

The EXIT command allows you to leave NETTRACE and return to the DCL prompt.

Syntax:

EXIT

This command has the same effect as Ctrl/Z.

11.3.8 HELP

This command displays on-line help information about NETTRACE.

Syntax:

HELP

This command has the same effect as Help.

11.3.9 NEXT

This command allows you to move (forward) through analyzed NETTRACE output one screen at time. See also the BACK command (Section 11.3.3).

Syntax:

NEXT

If you are running NETTRACE as a detached process, the output is directed to a file for later analysis. When the contents of the file are analyzed, the NEXT command can be used to turn off the (default) scroll mode and display the data one screen at a time. Each time you enter the command, the next screen of data is displayed.

During live tracing, the NEXT command has no effect until the BACK command has been used to suspend tracing and work back through the existing output. You can then use the NEXT command to display the next screen of data. However, once you reach the point where tracing was suspended, using the NEXT command has the effect of resuming live tracing.

11.3.10 REFRESH

The REFRESH command redraws the screen.

Syntax:

REFRESH

This command has the same effect as Ctrl/W.

11.3.11 SHOW

The SHOW commands allow you to display the trace collector or the key definitions for your keyboard. To display the trace collector, issue the following command.

SHOW TRACE_COLLECTOR [*node-spec*::][*process-name*][/[NO]FULL]

where

node-spec	is the gateway node name followed by the privileged username and password in quotes. For example, ALAS"SYSTEM SECRET".
process-name	Is a string that matches the data collecting process that you want to display. If you omit *process-name-string*, all trace collecting processes are displayed.

If you specify /FULL, the names of the lines or channels being traced are also displayed.

To display a key definition, issue the following command:

SHOW KEY [*key-name*]

where

key-name	Is the name of the key associated with the definition you want to display.

This command displays key definitions that you have defined using the DEFINE/KEY command or definitions that have been read from the NETTRACE key definition file. This file is pointed to by the logical name NETTRACE$KEY_INIT, and resides in SYS$LIBRARY:NETTRACE$KEY.INIT.

SHOW KEY qualifiers

The qualifiers available with the SHOW KEY command are as follows. (D) indicates a default qualifier.

/ALL

Displays all the key definitions. If you use the /ALL qualifier, do not specify a key name.

/FULL
(D) /NOFULL

Requests that all qualifiers that are associated with a definition are displayed.

/STATE=*state-name*
/NOSTATE=*state-name*

Specifies the name of a state table for which the specified key definitions are to be displayed. If you do not specify /STATE, the key definitions for all states are displayed.

11.3.12 SPAWN

The SPAWN command creates a subprocess of the current process.

Syntax:

SPAWN [*command-string*]

where

command-string	Is the command to be executed by the subprocess.

When the command completes, the subprocess terminates and control is returned to the parent process. If you use both the /INPUT qualifier and a command string, commands are obtained from the input files after the specified command string has been executed.

SPAWN qualifiers

The qualifiers available with the SPAWN command are as follows. (D) indicates a default qualifier.

/INPUT= *file-spec*

Specifies an input file that contains one or more DCL commands to be executed by the spawned subprocess. If you specify a command string with the SPAWN command and an input file with the /INPUT qualifier, the command string is processed before the input file. Once processing of the input file is complete, the subprocess is terminated.

(D) /LOGICAL_NAMES
/NOLOGICAL_NAMES

Determines whether the system passes logical names and logical name tables to the subprocess.

/OUTPUT=*file-spec*

Allows you to write the output from the SPAWN command to a specified file.

/PROCESS=*subprocess-name*

Specifies the name of the subprocess to be created. By default, a unique process name is assigned with the same base name as the parent process and a unique number. The default subprocess name format is username_n.

/PROMPT=*string*

Specifies the prompt string for DCL to use within the subprocess. By default, SPAWN copies the current prompt from the parent process.

/SYMBOLS
(D) /NOSYMBOLS

Determines whether the system passes DCL global and local symbols to the subprocess.

/WAIT
(D) /NOWAIT

Controls whether the system waits until the current subprocess is completed before allowing more commands to be issued by the parent process. The /NOWAIT qualifier allows you to issue new commands while the specified subprocess is running. When you use the /NOWAIT qualifier interactively, make sure that you use the /OUTPUT qualifier as well, so output from the subprocess is directed to a file rather than to your terminal. If you do not specify /OUTPUT, your terminal will be used by more than one process simultaneously.

11.3.13 START

The START command initiates a trace operation. The qualifiers you use with the command depend on whether you are performing live tracing or tracing from a detached process. The lists of qualifiers for live and detached tracing are in separate sections following the command syntax.

Syntax:

```
                              [[ { CHAN[.circuit-id] }  ]]
START[/LIVE] node-spec::      [[ { SDLC[.circuit-id] }  ]] /qualifiers
                              [[ PU[.pu-id]             ]]
```

where

node-spec	Is the name of the gateway node you want to trace and the privileged username and password associated with that gateway, in quotes. For example, ALAS"SYSTEM SECRET".
.circuit-id	Is the optional specification of a CHAN or SDLC circuit such as CHAN-0 or SDLC-1.
.pu-id	Is the optional name of a PU, for example, SNA-0.

The preceding syntax diagram indicates that you can specify either the CHAN or the SDLC tracepoint, depending on whether you are tracing a Gateway-CT or Gateway-ST. You can also specify the PU tracepoint along with CHAN or SDLC, or you can specify the PU tracepoint alone.

CHAN and SDLC circuit tracing record datalink frames, which include SDLC or CHAN link headers, TH (transmission header), and RH (request/response header) protocol headers and data.

PU tracing captures frames at the path control level which include TH and RH protocol headers and data. Note that session level tracing is specified as a qualifier to PU tracing.

START qualifiers (Live Tracing)

The following qualifiers are available when you use NETTRACE for live tracing. (D) indicates a default qualifier.

/BUFFER_SIZE=*n*

Specify the number and/or size (in bytes) of the internal buffers kept by NETTRACE. Increasing these values can help avoid losing trace records. The default for BUFFER_SIZE is 500.

/CAPTURE_SIZE=*n*

Specifies the maximum amount of data (in bytes) in each trace record captured. Decreasing the value of this qualifier can reduce the number of lost trace records. The default is 140.

/DATA=*(data_type,...)*
/NODATA

Allows you to specify the format of the output, which can be one or more of the following: HEXADECIMAL, OCTAL, ASCII, EBCDIC, or DECIMAL. In ASCII output, control characters are represented by " ^character"; for example Ctrl/M is represented by " ^M". The default is HEXADECIMAL.

/DISPLAY=*(display-option,...)*
/NODISPLAY

Allows you to specify the fields to appear in the analyzed output (see Table 11–2). The default display options are TIME, EVENT, and SIZE.

/EVENT=*(event-type,...)*

Allows you to select the events to include in the output, which can be one or more of the following: TRANSMIT, RECEIVE, or ALL. The default is ALL.

/MAXIMUM_BUFFERS=*n*

Specifies the maximum number of buffers TRACE uses to hold data. The default for BUFFER_SIZE is 500. If you are tracing a remote node, the default for MAXIMUM_BUFFERS is 3. If you are tracing the local node, the default for MAXIMUM_BUFFERS is 10.

/NAME=*channel-name*

Where *channel-name* is the name of the trace records that are to be analyzed. If you specify only part of the channel name, all records that start with this string are selected for analysis. Use the /DISPLAY=NAME option to display the channel name in the trace record. /NAME=* displays all the records.

/PAGE=*number*

Allows you to specify the number of lines generated per page. If the data is being output to a file, the default is 60.

/PROTOCOL=*(protocol-identifier,...)*
/NOPROTOCOL

Where *protocol-identifier* is one or more of the tracepoints. Use /NOPROTOCOL if you want the protocol headers to appear as part of the displayed data (that is, without being formatted). The default protocols depend on which tracepoints are being analyzed. /PROTOCOL (not normally used) allows you to specify which protocol is displayed. See Table 11–3 for tracepoints and the default protocol identifiers.

/REVERSE

/NOREVERSE (D)

If your terminal supports reverse video (and is set to DEC_CRT), specifying /REVERSE displays the received data in reverse video and displays the titles in bold.

/SAVE_BUFFER_SIZE=*n*

Where *n* is the number of screens of saved lines of data. These lines can be examined by using the NEXT and BACK commands. The default is 30.

/SELECT=*(protocol-identifier,...)*

/NOSELECT

Where *protocol-identifier* is one or more of the tracepoints. This qualifier allows you to specify the protocol headers to appear in the output. Any protocol headers that you select are removed from the analyzed data. The default is to display all the protocol headers (see Table 11–3).

/SESSION=(*session-range*,...)

/NOSESSION

Where *session-range* is a list of SNA session numbers and ranges. Use /NOSESSION to analyze trace records for all sessions. /SESSION allows you to specify which sessions to analyze. A session range can be a single value, range of values, or every session up to the maximum. For example, /SESSION=1,10-* specifies session 1 and all sessions greater than or equal to 10. This qualifier is valid for PU trace records only.

/TIME_OUT=*n* (units of 100ms)

Trace records are sent to the collector when a trace buffer is full or when a timeout occurs. The /TIME_OUT qualifier allows you to specify, in units of 100ms, the rate at which a timeout occurs. The default is 2.

/TRUNCATE

(D) /NOTRUNCATE

Determines whether data that will not fit on one line is truncated or is printed on the next line.

/WIDTH=*number*

Allows you to specify the number of columns generated per page. If the data is written to a file, the default is 132. If the data is displayed on your terminal, the default is the width of your terminal screen.

START qualifiers (Detached Tracing)

The following qualifiers are available when you run NETTRACE as a detached process. (D) indicates a default qualifier.

/BLOCKS=*n*

Specifies the approximate maximum size (in blocks) of each output file that is created. When the file reaches the maximum size, it is closed, and a new version of the file is opened. The default is 200.

/BUFFER_SIZE=*n*

Specify the number and/or size (in bytes) of the internal buffers kept by NETTRACE. Increasing these values can help avoid losing trace records. The default for BUFFER_SIZE is 500.

/CAPTURE_SIZE=*n*

Specifies the maximum amount of data (in bytes) in each trace record captured. Decreasing the value of this qualifier can reduce the number of lost trace records. The default is 140.

/MAXIMUM_BUFFERS=*n*

Specifies the maximum number of buffers TRACE uses to hold data. The default for MAXIMUM_BUFFERS is 3 if you are tracing a remote node, 10 if you are tracing the local node.

/OUTPUT=*filename*

Specifies a filename for your output. This file will contain binary trace records. The default is TRACE.DAT.

/PRIORITY=*n*

Specifies the base priority of the detached process. The default is 8.

/PROCESS_NAME=*process-name*

Allows you to specify an alternative name for the detached process. The default is *username$NETTRACE[_n]*. (If *username$NETTRACE[_n]* has more than 15 characters, the string NETTRACE is truncated to reduce the name to 15 characters.)

/TIME_OUT=*n* (units of 100ms)

Trace records are sent to the collector when a trace buffer is full or when a timeout occurs. The /TIME_OUT qualifier allows you to specify, in units of 100ms, the rate at which a timeout occurs. The default is 2.

/VERSION_LIMIT=*n*

Specifies the maximum number of versions of the output file kept. The default is 10.

11.3.14 STOP

The STOP command stops tracing.

Syntax:

STOP *[process-name]*

where

process-name	Is the collecting process you wish to stop.

To stop live tracing, issue the STOP command without specifying a process name.

To stop a detached trace collector, specify the name of the process you wish to stop. The default is *username$NETTRACE[_n]*. (If *username$NETTRACE[_n]* has more than 15 characters, the string NETTRACE is truncated to reduce the name to 15 characters.) You can display the names of the trace collectors by using the SHOW command.

11.4 Examples of NETTRACE Commands and Output

As described in Section 11.2 you can either analyze data as it is being collected and displayed on your terminal (LIVE tracing) or you can record binary data in a file for later analysis (detached process tracing). The following examples illustrate these two ways of using NETTRACE.

The first example starts a trace on the DECnet VAX Gateway node named ALACK with the privileged user name and password of SYSTEM and SECRET, respectively. The tracepoint specified is CHAN with the further specification that only data on the CHAN-0 circuit be captured.

```
$ TRACE
NETTRACE> START ALACK"SYSTEM SECRET"::CHAN.CHAN-0/LIVE
```

The next example illustrates detached process tracing where the output is sent to a file named SDLC.DAT. In this example the Gateway node is a DECnet/OSI node named VINAL, and the privileged user name and password are SYSTEM and SECRET, respectively. The tracepoint is SDLC and the command specifies that only the frames on the SDLC-0 circuit are recorded.

```
$ TRACE/SNA
NETTRACE> START VINAL"SYSTEM SECRET"::SDLC.SDLC-0/OUT=SDLC.DAT
```

The remainder of this section contains examples of trace output from each of the four tracepoints: CHAN, SDLC, PU, and session. The examples all share the following characteristics:

- The <-T symbol indicates a message transmitted by the Gateway; the R-> symbol indicates a message received by the Gateway.

- The data size field indicates the size of the message being traced, including the data link header (Address and Control fields).
- The TH and RH field displays the interpreted contents of an information frame's Transmission Header and Request/Response Header.

  Some TH and RH display lines are truncated because they are longer than the width of the field. You can increase the size of the field by specifying other display qualifier options (for example, /DISPLAY=NOSIZE).
- The data field shows the actual data in a frame and is displayed according to the format specified in the /DATA option in START or ANALYZE commands. In the following examples, some of the data fields have been truncated in order to keep the examples short.

11.4.1 CHAN Tracepoint Example

The following commands

```
$ TRACE START ALACK"SYSTEM SECRET"::CHAN/OUTPUT=CHAN.DAT
$ TRACE STOP
$ TRACE ANALYZE CHAN.DAT/OUTPUT=CHAN.TXT/DISPLAY=NAME
```

produce the output shown in Example 11–1.

Example 11–1 Sample Output of a CHAN Tracepoint

```
Trace started on 17-OCT-1989 22:55:54.58  Analyzed on 24-OCT-1989 13:39:15.53
Trace File [DILLINGHAM]CHAN.DAT;9       Output File [DILLINGHAM]CHAN.TXT;1
-----------+----+----+---------------+<------- CHAN Info ----->+<------------------------------>+---------------
    Time   |Evnt|Data|Name of        |                         |               TH               |Data
hh mm ss cc|    |Size|Line or Channel|Command         Status   |               RH               |
-----------+----+----+---------------+<----------------------->+<------------------------------>+---------------
22:55:54.58| <-T|   0|CHAN-0         |                A        |
22:55:54.58|R-> |   0|CHAN-0         |Read Start 0    CE       |
22:55:54.58| <-T|   0|CHAN-0         |                DE       |
22:55:54.58| <-T|  16|CHAN-0         |Read          CE/DE/UE/A |FID2,OS,DAF=00,OAF=FA,SNF=0004 | 01 06 04
                                                                 +RSP,FMD,FI,BCI,ECI,DR1I
22:55:54.59|R-> |   0|CHAN-1         |Read Start 0    CE       |
22:55:54.59| <-T|   0|CHAN-1         |                DE/UE    |
23:11:08.58| <-T|   0|CHAN-1         |                A        |
23:11:08.58|R-> |   0|CHAN-1         |Read Start 1    CE       |
23:11:08.58| <-T|   0|CHAN-1         |                DE       |
23:11:08.58| <-T|  27|CHAN-1         |Read          CE/DE/UE/A |FID2,OS,DAF=00,OAF=FA,SNF=0003 | 01 06 81 00 E7
                                                                 RQ,FMD,FI,BCI,ECI,DR1I           D5 D1 C5 E2 C3
                                                                                                  E2 00 00 00
23:11:08.59|R-> |   0|CHAN-0         |Read Start 1    CE       |
23:11:08.59| <-T|   0|CHAN-0         |                DE/UE    |
23:11:08.60|R-> |   0|CHAN-0         |Write Start 0   CE       |
23:11:08.60| <-T|   0|CHAN-0         |                DE       |
23:11:08.61|R-> |  16|CHAN-0         |Write Break     CE       |FID2,OS,DAF=FA,OAF=00,SNF=0003 | 01 06 81
                                                                 +RSP,FMD,FI,BCI,ECI,DR1I
23:11:08.61| <-T|   0|CHAN-0         |                DE       |
```

LKG-5303-91R

The abbreviations in the CHAN Info Status field are explained in Table 12–4.

11.4.2 SDLC Tracepoint Example

The command

```
$ TRACE START/LIVE/DISPLAY=NAME CLIFF"SYSTEM SECRET"::SDLC
```

produces the output shown in Example 11–2.

Example 11–2 Sample Output of an SDLC Tracepoint

```
-----------+----+---------------+-----+<----------SDLC--------->+<----------------------------->+---------------
    Time   |Evnt|Name of        |Data |M  A  T                  |              TH               |Data
hh mm ss cc|    |Line or Channel|Size |d  d  y  CMD  P/F Nr  Ns |              RH               |
-----------+----+---------------+-----+<----------------------->+<----------------------------->+---------------
10:49:46.87| <-T|SDLC-0         |    3|X 40  S  RR    0 112     |
10:49:46.87| <-T|SDLC-0         |   42|X 40  I  INFO  1 112  19 |FID2,OS,DAF=01,OAF=08,SNF=0001 | 7D D5 F1 11 D4
                                                                 RQ,FMD,BCI,ECI,DR1I,BBI,CDI      5C 84 89 93 93
10:49:46.87|R-> |SDLC-1         |    2|N 40  S  RR    1   2     |
10:49:46.87| <-T|SDLC-1         |    2|N 40  S  RR    1   0     |
10:49:46.96|R-> |SDLC-0         |    3|X 40  S  RR    1  20     |
10:49:46.97| <-T|SDLC-0         |    3|X 40  S  RR    1 112     |
10:49:46.99|R-> |SDLC-0         |   12|X 40  I  INFO  0  20 112 |FID2,OS,DAF=08,OAF=01,SNF=0001 |
                                                                 +RSP,FMD,BCI,ECI,DR1I,DR2I
10:49:47.01|R-> |SDLC-0         |   63|X 40  I  INFO  0  20 113 |FID2,OS,DAF=08,OAF=01,SNF=0003 | F1 C2 11 40 7B
                                                                 RQ,FMD,BCI,ECI,DR1I,EBI          F1 F0 61 F2 F4
                                                                                                  61 F8 F9 40 F1
10:49:47.01|R-> |SDLC-1         |  264|N 40  I  INFO  0   3   7 |FID2,MS,DAF=02,OAF=01,SNF=005B | E3 C5 D9 61 C1
                                                                                                  40 C1 D5 C4 11
                                                                                                  C4 C1 1D F0 D9
10:49:47.07|R-> |SDLC-0         |    3|X 40  S  RR    1  20     |
10:49:47.07| <-T|SDLC-0         |    3|X 40  S  RR    0 114     |
10:49:47.07| <-T|SDLC-0         |   12|X 40  I  INFO  1 114  20 |FID2,OS,DAF=01,OAF=08,SNF=0003 |
                                                                 +RSP,FMD,BCI,ECI,DR1I
```

LKG-5304-91R

For information on how to interpret the data in the SDLC frame (the fifth column in the NETTRACE output) see Section 11.4. The 'X' and 'N' in the first column of the SDLC field indicate extended response mode and normal response mode, respectively. The headings in the SDLC field — Md, Ad, and Ty (shown vertically) — stand for Mode, Address, and Type of Frame, respectively.

11.4.3 PU Tracepoint Example

The commands

```
$ TRACE START ALACK"SYSTEM SECRET"::PU/OUT=PU6.LIS
$ TRACE STOP
$ TRACE ANALYZE/DATA=EBCDIC/OUT=PU6.TXT PU6.LIS
```

produce the output shown in Example 11–3.

Example 11–3 Sample Output of a PU Tracepoint

```
Trace started on 26-OCT-1989 12:51:18.85  Analyzed on 26-OCT-1989 12:52:13.86
Trace File USER2$:[DILLINGHAM]PU6.LIS;1 Output File USER2$:[DILLINGHAM]PU6.TXT;1
-----------+----+-----+<---------------------------------->+-----------------------
    Time   |Evnt|Data |              TH                    |Data
hh mm ss cc|    |Size |              RH                    |
-----------+----+-----+<---------------------------------->+-----------------------
12:51:36.25|R-> |  470|FID2,OS,DAF=01,OAF=01,SNF=0010      |1B. #10/26/89 12:52:37.-A
                       RQ,FMD,BCI,ECI,DR1I,BBI,EBI          .A&.B-..A&.-* CNM02
                                                            LOGONT.B-.C0..B-.-- CN
                                                            M02
                                                            DSI020I OPERATOR DILLING
                                                            LOGGED ON FROM
                                                            TERMINAL H01E100A
12:51:36.34| <-T|    9|FID2,OS,DAF=01,OAF=01,SNF=0010      |
                       +RSP,FMD,BCI,ECI,DR1I
12:51:36.35|R-> |  191|FID2,OS,DAF=01,OAF=01,SNF=0011      |1B. #10/26/89 12:52:37.-A
                       RQ,FMD,BCI,ECI,DR1I,PI,BBI,EBI       .[ ..&..[ .-.[<+
                                                            : "DISPFK" TO SEE YOUR PF
                                                            KEY SETTINGS..&.<-...&.-C
                                                            CNM02    CNM357I PFKDE
12:51:36.40| <-T|    9|FID2,OS,DAF=01,OAF=01,SNF=0000      |
                       +RSP,FMD,BCI,ECI,PI
12:51:36.40| <-T|    9|FID2,OS,DAF=01,OAF=01,SNF=0011      |
                       +RSP,FMD,BCI,ECI,DR1I
12:51:41.23| <-T|   21|FID2,OS,DAF=01,OAF=01,SNF=0002      |'*6.*0logoff
                       RQ,FMD,BCI,ECI,DR1I,ERI,BBI,CDI
12:51:41.25|R-> |   57|FID2,OS,DAF=01,OAF=01,SNF=0012      |1B. #10/26/89 12:52:42.-A
                                                            .*0.  .$Z.*?....A(A
                       RQ,FMD,BCI,ECI,DR1I,EBI
12:51:41.26|R-> |   11|FID2,OS,EFI,DAF=01,OAF=01,SNF=39E0  |..
                       RQ,SC,FI,BCI,ECI,DR1I
12:51:41.30| <-T|   10|FID2,OS,EFI,DAF=01,OAF=01,SNF=39E0  |.
                       +RSP,SC,FI,BCI,ECI,DR1I
12:51:41.36|R-> | 1306|FID2,OS,DAF=01,OAF=00,SNF=001A      |----------------------
                       RQ,FMD,BCI,ECI,DR1I                  D I G I T A L
                                                            E Q U I P M E N T
                                                            C O R P O R A T I O N
                                                            ----------------------
                                                            IBM INTERCONNECT CENTER
```

LKG-5305-91R

11.4.4 Session Tracepoint Example

The command

```
$ TRACE START ALACK"SYSTEM SECRET"::PU.SNA-0/SESSION=2/LIVE
```

produces the output shown in Example 11–4.

Example 11–4 Sample Output of a Session Tracepoint

```
-----------+----+-----+<--------------------------------->+----------------------------------------
    Time   |Evnt|Data |               TH                  |Data
hh mm ss cc|    |Size |               RH                  |
-----------+----+-----+<--------------------------------->+----------------------------------------
10:10:04.48|R-> |   17|FID2,OS,DAF=02,OAF=01,SNF=000F      | F1 C2 11 C1 4D C1 40 40
                      RQ,FMD,BCI,ECI,DR1I,BBI,EBI
10:10:04.50| <-T|    9|FID2,OS,DAF=01,OAF=02,SNF=000F      |
                      +RSP,FMD,BCI,ECI,DR1I
10:10:04.61|R-> |  191|FID2,OS,DAF=02,OAF=01,SNF=0010      | F1 C2 11 40 7B F1 F0 61 F3 F1 61 F8 F9
                      RQ,FMD,BCI,ECI,DR1I,BBI,EBI           40 F1 F0 7A F0 F9 7A F1 F5 1D 60
10:10:04.67| <-T|    9|FID2,OS,DAF=01,OAF=02,SNF=0010      |
                      +RSP,FMD,BCI,ECI,DR1I
10:10:09.54| <-T|   21|FID2,OS,DAF=01,OAF=02,SNF=0002      | 7D 5C F6 11 5C F0 93 96 87 96 86 86
                      RQ,FMD,BCI,ECI,DR1I,BBI,CDI
10:10:09.62|R-> |    9|FID2,OS,DAF=02,OAF=01,SNF=0002      |
                      +RSP,FMD,BCI,ECI,DR1I,DR2I
10:10:09.63|R-> |   57|FID2,OS,DAF=02,OAF=01,SNF=0011      | F1 C2 11 40 7B F1 F0 61 F3 F1 61 F8 F9
                      RQ,FMD,BCI,ECI,DR1I,PI,EBI            40 F1 F0 7A F0 F9 7A F2 F0 1D 60
10:10:09.64|R-> |   11|FID2,OS,EFI,DAF=02,OAF=01,SNF=3459  | 32 01
                      RQ,SC,FI,BCI,ECI,DR1I
10:10:11.63| <-T|   10|FID2,OS,EFI,DAF=01,OAF=02,SNF=3459  | 32
                      +RSP,SC,FI,BCI,ECI,DR1I
10:10:11.69|R-> | 1306|FID2,OS,DAF=02,OAF=00,SNF=0004      | 60 60 60 60 60 60 60 60 60 60 60 40 C4
                      RQ,FMD,BCI,ECI,DR1I                   40 C9 40 C7 40 C9 40 E3 40 C1 40
```

LKG–5306–91R

11.5 NETTRACE Messages

NETTRACE messages are produced either by the NETTRACE utility or by the detached NETTRACE collecting process. Messages produced by the detached process are displayed as OPCOM messages. The detached NETTRACE collector always produces an OPCOM message just before it exits. The message explains why NETTRACE is exiting.

Appendix C describes each NETTRACE message, reason for the message, and a possible remedy for the problem, if appropriate. In addition to these messages, the analyzed trace file contains entries of the form:

n trace records lost

Trace records are lost when they are being produced more quickly than they can be written to the terminal or to a file (although they are more likely to occur during live tracing). To avoid losing trace records, use the START command qualifiers as follows:

1. Specify a smaller value for CAPTURE_SIZE.
2. Specify a larger value for BUFFER_SIZE.
3. Specify a larger value for MAXIMUM_BUFFERS.

12
Using SNATRACE

SNATRACE, like NETTRACE, is a command-driven utility that you use to monitor, record, and analyze the flow of data between a DECnet SNA Gateway and an SNA network. SNATRACE and NETTRACE are similar in basic function, however, NETTRACE is a newer utility and provides additional features. See Chapter 11 for information on NETTRACE.

12.1 Starting SNATRACE

SNATRACE is an OpenVMS host-based trace utility. It works with a trace server on any version of the Gateway systems to collect and analyze the flow of data between the Gateway and the IBM system. (When using SNATRACE on Gateway V1.3 or earlier, you must specify the qualifier /OLD_VERSION with the TRACE command.)

To start SNATRACE enter the following command:

```
$ RUN SYS$SYSTEM:SNATRACE
```

SNATRACE then prompts you for a command:

```
SNATRACE>
```

You can also create a DCL foreign command that invokes the utility. Enter the following at the DCL level or place a similar assignment in your LOGIN.COM file:

```
$ SNATRACE == "$SNATRACE"
```

Refer to the *VMS Mini-Reference* for more information on DCL foreign commands.

Once you have defined a foreign command to invoke SNATRACE, you can enter a complete SNATRACE command on the DCL command line. The following example uses a symbol to invoke SNATRACE and then specifies an SNATRACE subcommand at the DCL level.

```
$ SNATRACE ANALYZE GORDON.DAT/OUTPUT=SNAGWY.LIS
```

In this example, SNATRACE analyzes the binary trace data in the file GORDON.DAT and produces trace output in the file SNAGWY.LIS.

12.2 How SNATRACE Works

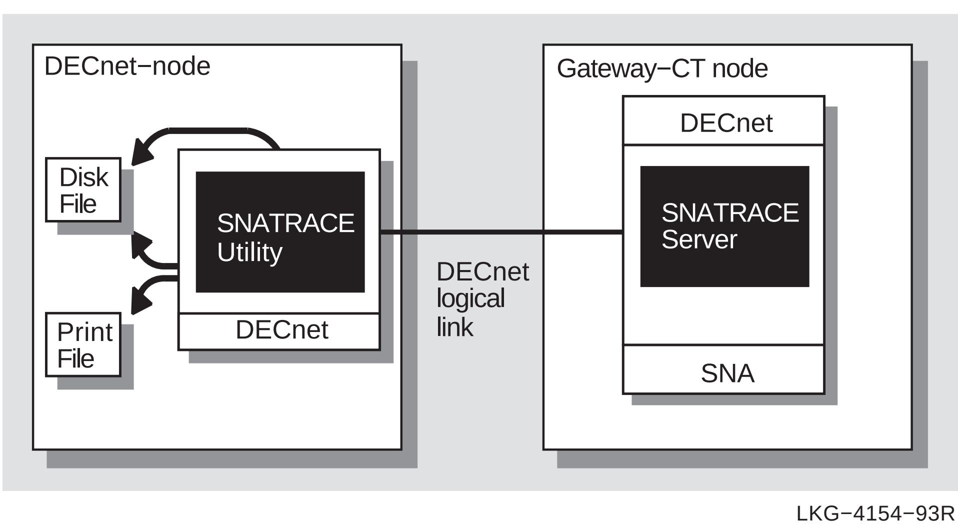
Once the SNATRACE image is running, you can enter the TRACE command with appropriate access control information. In the TRACE command, you identify the following:

- The Gateway node and circuit or PU on which the trace is to be run
- The level of trace you want
- The name and characteristics of a file to contain the trace data

The TRACE command instructs the trace server on the Gateway to collect trace data. The trace server sends the data it collects to SNATRACE, where it can be analyzed and formatted immediately or stored in a binary file.

Figure 12–1 shows the system components used for a trace.

Figure 12–1 System Components Used by SNATRACE



12.2.1 Trace Levels

You can use SNATRACE to obtain trace data that shows protocol sequences exchanged on the following three levels:

- Circuit level
- Physical unit level
- Session level

12.2.1.1 Circuit Level Trace — SDLC Circuits

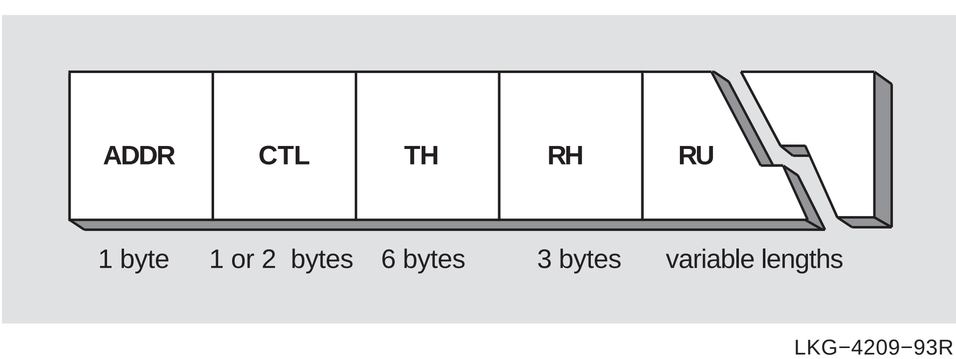
This trace level retrieves messages transmitted to or received from the IBM system as they are exchanged between the SDLC module and the device driver in the Gateway node.

Use this trace level when you have an SDLC problem. Symptoms of an SDLC problem include the following:

- The line is not being polled.
- The circuit to IBM does not initialize properly. Improper initialization occurs when the circuit starts but changes to the ON–STARTING state, or when the circuit state is ON–DISCONNECTED.
- You receive duplicate messages

Figure 12–2 illustrates the contents of a circuit level trace message when your SDLC circuit is set to normal response mode (modulo 8).

Figure 12–2 SDLC Circuit Level Trace Message



If your circuit is set to extended response mode (modulo 128), the SDLC circuit level trace message has 1 or 2 bytes for the control field (CTL).

Messages retrieved at this level consist of an address field (ADDR), a control field (CTL), and a data field (TH/RH/RU). The values for the ADDR and CTL fields are taken from the link header (LH) of the SDLC frame in which the message is enclosed. Table 12–1 and Table 12–2 explain these values. (The table shows the values in binary form. To use the table, convert the hexadecimal values the ADDR and CTL fields yield to binary form.)

TH/RH/RU stands for the transmission header, request/response header and request/response unit that represent the data field. If an SDLC information frame (I–frame) is being transmitted or received by the Gateway-ST, the data consists of a TH/RH/RU element. If an SDLC supervisory or nonsequenced frame is being transmitted, the data, if any, is associated with the SDLC command or response in the control field. Information on TH/RH/RU units and the bits that comprise them is available in the *DECnet SNA Application Programming Interface for OpenVMS Guide*. Refer to the *Systems Network Architecture Formats* for a complete discussion of this subject.

12.2.1.2 Circuit Level Trace — Channel-Attached Circuits

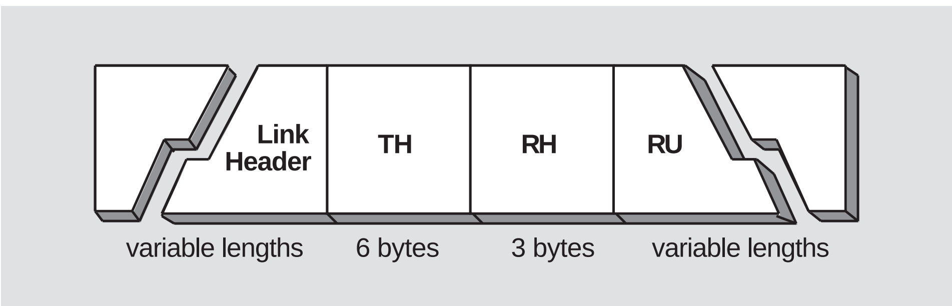
This trace level retrieves messages transmitted to or received from the IBM system as they are exchanged between the channel module and the device driver in the Gateway node.

Use this trace level when you have a channel problem. Symptoms of a channel problem include the following:

- The channel-attached circuit to the IBM does not initialize properly. Improper initialization occurs when the circuit starts but changes to the ON–STARTING state, or when the circuit state is ON–DISCONNECTED.
- You receive duplicate messages.
- You suspect that messages are being lost.
- You receive garbled messages.

Figure 12–3 illustrates the contents of a circuit level trace when you are using channel-attached circuits.

Figure 12–3 Channel-Attached Circuit Level Trace Message



Messages retrieved at the channel-attached circuit level consist of a command field (CMD), a status field (STS), and a data field (LH/TH/RH/RU).

The values for the CMD field indicate the type of channel command being executed over the channel-attached circuit. The values for the STS field indicate the status being returned to the channel in response to this command. Table 12–3 and Table 12–4 explain the CMD and STS field values.

LH/TH/RH/RU stands for the link header, transmission header, request/response header, and request/response unit that comprise the data field.

12.2.1.3 Physical Unit Level Trace

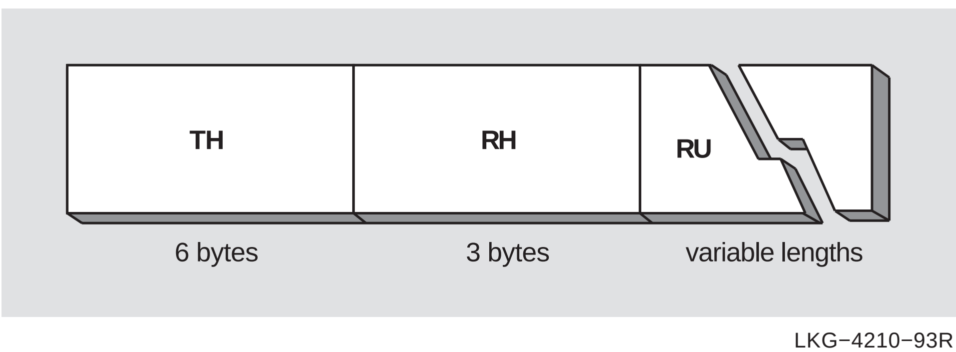
This trace level retrieves all messages handled by a specified physical unit (PU) as they are exchanged between the SNA modules and the SDLC or channel module in the Gateway node. This includes all messages on all the LUs controlled by the PU.

The physical unit level trace is used by default, if you do not specify a level. Select this trace level when any of the following occurs:

- You cannot establish a connection between the Gateway and IBM
- You establish a connection between the Gateway and IBM but it fails repeatedly
- You receive a message indicating that there is a protocol problem on the PU level

Figure 12–4 shows the contents of a PU level trace message.

Figure 12–4 PU Level Trace Message



Messages returned from a PU level trace are formatted in the same way as a circuit level trace; however, only the data field (TH/RH/RU) is returned.

12.2.1.4 Session Level Trace

This trace level retrieves all messages transmitted to or received from an IBM system in a specified session as they are exchanged between the SNA module and the data link module in the Gateway node.

Select this trace level when any of the previously listed symptoms are present and also when protocol problems occur in a specific session or particular application.

The information retrieved for a session level trace is in the same format as the information for a PU level trace. See Figure 12–4 for the contents of a session level trace message.

12.3 SNATRACE Commands

SNATRACE has five commands:

- TRACE
- ANALYZE
- HELP
- SHOW
- EXIT

These commands are explained in the following sections.

12.3.1 TRACE

The TRACE command initiates a protocol trace operation. When you enter the TRACE command, you must also enter a node specification (a node identification followed by access control information) and a circuit or PU identification to identify the level of the trace.

Command qualifiers for the TRACE command enable you to specify the type of trace and the particular characteristics of the trace file, such as file size and the number of entries logged. If you specify a value larger than that allowed for a qualifier, SNATRACE uses the maximum allowable value and does not issue an error message. Trace data is returned to the SNATRACE program, which then records the information in a file or displays it.

Note

If you enter Ctrl/C after issuing the TRACE command, SNATRACE terminates the operation in an orderly manner.

Syntax:

TRACE *node-spec*:: $\left\{ \begin{array}{l} \textit{circuit-id} \\ \textit{pu-id} \\ \textit{session-id} \end{array} \right\}$ /*qualifiers*

where

node-spec — Is the node name of the Gateway followed by a privileged username and password (you must provide these with the TRACE command). You define the privileged username and password in the Gateway configuration procedure, as described in *DECnet SNA Gateway for Synchronous Transport Installation* or *DECnet SNA Gateway for Channel Transport Installation*.

circuit-id — Is the identification of the circuit to be used. When used with an SDLC circuit level trace, this parameter is a string in the form SDLC-*nnn*, where *nnn* identifies a particular SDLC circuit connecting with the IBM system. When used with a channel-attached circuit level trace, this parameter is a string in the form CHAN-*nnn*, where *nnn* identifies a particular channel-attached circuit connecting with the IBM system.

pu-id — Is a string in the form SNA-*nnn*, where *nnn* identifies a particular PU connecting with the IBM system. For a PU level trace, specify *pu-id* with the /PU qualifier.

session-id	Is a string in the form SNA-*n.m*, where SNA-*n* is a pu-id and *m* is a session address. For a session level trace, specify *session-id* with the /SESSION qualifier.

If you enter the TRACE command without specifying any parameters or qualifiers, you are prompted as follows:

```
SNATRACE> TRACE Return
_Node id:
_Component id:
```

In response to the component id prompt, enter either a circuit identification, session identification, or a PU identification and the relevant qualifiers.

TRACE qualifiers

The qualifiers to the TRACE command follow:

/ANALYZE
/NOANALYZE (D)

The /ANALYZE qualifier formats and records trace data as ASCII data (data that is not analyzed is recorded in binary form). Because this qualifier increases the use of system resources during a trace operation, analyze the data using the ANALYZE command instead of using this qualifier. If you do not specify /OUTPUT with /ANALYZE, the analyzed data is written to SYS$OUTPUT.

/NOANALYZE is the default.

/BUFFERS=*n*

This qualifier lets you specify the number of buffers are available on the Gateway node to receive trace data. If you receive many "trace data lost" errors, use the /BUFFERS qualifier to allocate more buffers.

The default is 10 buffers.

Generally, you can run SNATRACE with the default values for the /BUFFER and /SIZE qualifiers. However, if your process hangs when you run SNATRACE with these defaults, increase the buffered I/O quotas for your process. The quotas must exceed the values shown:

- BIOLM > (number of BUFFERS) + 2
- BYTLM > BIOLM times (value for SIZE)
- ASTLM > (number of BUFFERS) + 2

The values for BIOLM and BYTLM are defined for your user id in the SYSUAF file. (Typical values are BIOLM = 6, BYTLM = 20480, and ASTLM = 10.) Refer to the appendix on system configuration guidelines in the *VMS Authorize Utility Reference Manual* for more information.

/CHARACTER_SET

You can use this qualifier to identify the input file containing the EBCDIC to Digital Multinational Character Set (DMCS) translation table and the DMCS-to-EBCDIC translation table. Refer to Appendix A for additional information.

If /CHARACTER_SET is not specified, the SNATRACE software uses the translation table named by the default file SYS$LIBRARY:SNATRADEF.TBL, if it exists. Otherwise, it uses its own table.

If you specify /CHARACTER_SET=ORIGINAL, SNATRACE uses its own table even if the SNATRADEF.TBL file is present.

Wildcard characters are not allowed in the file specification.

/CIRCUIT

You can use this qualifier to specify circuit level trace for either SDLC circuits or channel-attached circuits. Do not use this qualifier with the /PU or /SESSION qualifiers.

/ENTRIES=*n*

This qualifier closes the trace output file and opens a new file after a specified number of trace data entries has been received. The value of *n* must be between 1 and 32767. If you do not specify this qualifier, an entry counter is not maintained, and trace data is transmitted to the specified file until a significant event, such as pressing Ctrl/C, halts SNATRACE.

/OLD_VERSION

This qualifier forces the SNATRACE software to use the V1.0 protocol. Use /OLD_VERSION to trace DECnet SNA Gateway Version 1.3 or earlier.

/OUTPUT=*file-spec*

This qualifier identifies a file on a DECnet node that contains the trace data sent by the Gateway node.

If you do not specify a file name and extension for the *file-spec* value, SNATRACE.DAT is used by default. (If you specify /ANALYZE in the command line, .LIS is the default file extension.) If you do not specify a directory and device name for the file, the TRACE command uses your current default device and directory.

You cannot enter wildcard characters in the file specification.

/PU
This qualifier specifies a physical unit trace. Do not use this qualifier with the /CIRCUIT or /SESSION qualifiers. If you do not specify a trace level in the command line, a PU level trace is performed by default.

/SESSION
This qualifier specifies a trace of the logical unit. Do not use this qualifier with the /CIRCUIT or /PU qualifiers.

/SIZE=*n*
This qualifier indicates the maximum number of RU data bytes returned for each trace entry. (LH, TH, and RH header sizes are added to this number where appropriate.) This qualifier reduces the amount of trace data transmitted over the logical link and written into the data file. The value of *n* must be between 0 and 32767. The default is 256.

/VERSION_LIMIT=*n*
This qualifier specifies the maximum number of trace output files. Each time an output file is opened, the files are purged. The value of *n* must be between 2 and 32767.

/WIDE
/NOWIDE (D)
This qualifier is used with /ANALYZE to specify the width of the formatted trace data. If you specify /WIDE, the output data is arranged in a width of 132 columns.

/NOWIDE is the default. If you specify /NOWIDE or omit this qualifier entirely, the width is set at 80 columns.

12.3.2 ANALYZE

The ANALYZE command copies the binary information in a specified trace data file to a new file in ASCII format to help you interpret the data.

Syntax:

ANALYZE *file-spec*

where

file-spec	Identifies the trace data file to be analyzed. This is the file created by the TRACE command. The *file-spec* value is a standard OpenVMS file specification.

If you do not specify a file type for this file, .DAT is the default. You cannot use wildcard characters in the file specification.

If you do not specify a file-spec for the ANALYZE command, you are prompted for one as follows:

```
SNATRACE> ANALYZE Return
_File:
```

ANALYZE qualifiers

The qualifiers to the ANALYZE command are the following:

/OUTPUT=*file-spec*

This qualifier identifies the output file created to store the results of the analysis. The default file type is .LIS. If you do not enter a file name, the output is sent to a file that has the same name as the input file and a file type of .LIS. If you do not specify the /OUTPUT qualifier, the output is sent to SYS$OUTPUT. You cannot use wild-card characters in the file specification.

/WIDE
/NOWIDE (D)

This qualifier specifies the width of formatted trace data. If you specify /WIDE, the output data is arranged in a width of 132 columns.

/NOWIDE is the default. If you specify /NOWIDE or omit this qualifier entirely, the width is 80 columns.

12.3.3 HELP

The HELP command gives you on-line information about the SNATRACE utility.

Syntax:

HELP [*keyword*...]

where

keyword...	Specifies one or more subjects about which you are requesting help. You can get help on the TRACE, ANALYZE, EXIT, and HELP commands and their qualifiers.

12.3.4 SHOW

The SHOW command displays information about the keyword you specify.

Syntax:

SHOW [*keyword*]

where

keyword	Specifies the information you want to display. Currently, the only keyword is VERSION. The SHOW VERSION command displays the version number of SNATRACE that you are running.

12.3.5 EXIT

The EXIT command stops SNATRACE and returns control to the OpenVMS command interpreter.

The following example stops SNATRACE:

```
SNATRACE> EXIT
$
```

You can also press Ctrl/Z to exit from SNATRACE.

12.3.6 SNATRACE Example

The following example shows the command syntax for a session level trace performed on session number 2 on PU SNA-0.

The output file created to contain the trace data is called EXAMPLE.DAT. To stop the trace, press Ctrl/C. When the trace has ended, the SNATRACE utility analyzes the file EXAMPLE.DAT, and the formatted results are stored in a file called EXAMPLE.LIS.

```
$ RUN SYS$SYSTEM:SNATRACE
SNATRACE> TRACE NODEG"SYSTEM SECRET" SNA-0.2 /OUTPUT=EXAMPLE.DAT/SESSION
%SNATRACE-I-BEGIN, beginning trace at 25-JUN-1988 16:37:22.40

Ctrl/C

%SNATRACE-I-ENDED, trace ended at 25-JUN-1988 17:15:30.70
SNATRACE> ANALYZE/OUTPUT=EXAMPLE.LIS EXAMPLE.DAT
SNATRACE> EXIT
```

12.4 Interpreting SNATRACE Messages

Every trace message that you retrieve contains a message-type value. There are two possible message-type values that you see:

- R—Indicates that the message was received by the Gateway.
- T—Indicates that the message was transmitted by the Gateway.
- C—Indicates a request to perform a control function (applies only to DECSA or DX24 Gateways).
- D—Indicates that the requested control function has completed (applies only to DECSA or DX24 Gateways).

12.4.1 SDLC Circuit Level Trace

An SDLC circuit level trace returns values for an address field (ADDR) and a control field (CTL). See Table 12–1 and Table 12–2 for information about these values. All levels of trace messages contain values for TH/RH/RU elements. See the appropriate IBM documentation for information on these values.

The sample SNATRACE messages in Example 12–1 show only the messages returned for an SDLC circuit level trace. Messages returned for PU level and session level traces are formatted in the same way as the circuit level trace, except that the ADDR and CTL fields are not returned.

Each SNATRACE message in Example 12–1 is marked by a callout number. A corresponding number and explanatory text for that item follow the example.

Example 12–1 SNATRACE Output for SDLC Circuit

```
$ RUN SYS$SYSTEM:SNATRACE

SNATRACE> TRACE/ANALYZE NODEG"SYSTEM SECRET":: SDLC-0/CIRCUIT/SIZE=267

  SNATRACE Version 2.1    Circuit Trace    23-JUN-1988 14:37:51.93
  Gateway NODEG  Circuit SDLC-0  Type SDLC-Normal
  (Protocol Version = 2.0.0, Buffering level = 10, Data size = 267)
                           .
                           .
                           .
```

(continued on next page)

Example 12–1 (Cont.) SNATRACE Output for SDLC Circuit

```
1  R 11:39:49.46 Ctl=51         Addr=40                (005,00000001)
                 TH=none        RH=none   RU=0. bytes
2  T 11:39:49.46 Ctl=94         Addr=40                (005,00000000)
                 TH=2C0000020003 RH=0B8000 RU=22. bytes
                 FID2,OS,DAF=00,OAF=02,SNF=0003
                 RQ,FMD,FI,BCI,ECI,DR1I
                 0106 8100 C4E2 C9D3 C7D4 D6C4 F305 C3D5 :.\a.DSILGMOD3.CN
                 D4F0 F200 0000                          : M02...
                              .
                              .
                              .
3  R 11:39:50.82 Ctl=71         Addr=40                (005,00000001)
                 TH=none        RH=none   RU=0. bytes
4  T 11:39:50.82 Ctl=D1         Addr=40                (005,00000000)
                 TH=none        RH=none   RU=0. bytes
                              .
                              .
                              .
5  R 11:39:52.06 Ctl=71         Addr=40                (005,00000001)
                 TH=none        RH=none   RU=0. bytes
6  T 11:39:52.06 Ctl=C1         Addr=40                (005,00000000)
                 TH=none        RH=none   RU=0. bytes
                              .
                              .
                              .
7  R 11:40:02.77 Ctl=04         Addr=40                (005,00000001)
                 TH=2C0002000004 RH=8B8000   RU=3. bytes
                 FID2,OS,DAF=02,OAF=00,SNF=0004
                 +RSP,FMD,FI,BCI,ECI,DR1I
                 0106 83                                 : .\c
8  R 11:40:02.81 Ctl=11         Addr=40                (005,00000001)
                 TH=none        RH=none   RU=0. bytes
9  T 11:40:02.81 Ctl=A1         Addr=40                (005,00000001)
                 TH=none        RH=none   RU=0. bytes
```

The following statements explain the numbered items in Example 12–1.

1 This is a one-line polling message. The first character on this line tells what type of message it is. The value R indicates that the polling message has been received by the Gateway-ST. The field following the message-type value shows a CTL value of 51. This is a hexadecimal value, as is the value returned in the ADDR field. (You must convert these hexadecimal values to binary form before you can interpret their meaning. See Table 12–1 or Table 12–2.)

A polling message does not have TH, RH, or RU elements as shown by the null values returned for each of these fields. The hexadecimal values in parentheses that constitute the final field in this message are function and status values. These values are not useful for general trace purposes.

2 The second message consists of six lines. The first field displayed shows a message-type value of T indicating a transmitted message. The next two fields contain the CTL value and the ADDR value. Following the ADDR field is the TH/RH/RU element. The TH field is six bytes in length and contains the hexadecimal value 2C0000020003. The RH field is three bytes long and contains the hexadecimal value 0B8000. The RU field is variable in length. In this example, the RU field contains 22 (decimal) bytes of data. (Function and status fields complete the first line of this message.)

Note

An asterisk (*) immediately following the word bytes in the RU field length indicates that the complete message was not displayed. For example, if you specify a /SIZE parameter of 64 bytes, and 256 bytes are actually transmitted, only 64 bytes of data are displayed.

On the third line of the message, TH data is displayed beginning with the term FID2. This information is explained in the *Systems Network Architecture Technical Overview*.

The fourth line shows the RH data, beginning with the term RQ. This information is also explained in the IBM documentation.

The final two lines in the message, beginning with the indented value 0106, represent the RU data. The RU data is printed before the colon in hexadecimal values and after the colon in EBCDIC values.

Note

Some messages do not contain data in the TH, RH, or RU fields.

---

3 The next message is a polling message, in the same format as message 1.

4 The last message in this series is a polling response transmitted by the Gateway-ST to the IBM system.

5 This message is a polling message, in the same format as message 1.

6 This message is a polling response from the Gateway-ST to the IBM system, as in message 4.

7 This message has been received by the Gateway-ST from the IBM system, in the same format as message 2.

8 This message is a polling message received by the Gateway-ST from the IBM system, as in message 1.

9 This message is a polling response from the Gateway-ST to the IBM system, as in message 4.

12.4.2 ADDR and CTL Fields in Messages

Table 12–1 and Table 12–2 contain explanations of the bits returned in the ADDR and CTL fields of SDLC circuit level trace messages. Table 12–1 explains the ADDR and CTL fields for normal response (modulo 8), and Table 12–2 explains the fields for extended response mode (modulo 128). Be sure to convert the hexadecimal values returned in the message to binary values before using this table.

Note

All command and response codes are documented according to IBM numbering conventions; that is, bit 0 at the far left of the message is the most significant bit. This is the reverse of Digital numbering conventions. Refer to *System Network Architecture Formats* for a complete description of the SDLC protocol.

Table 12–1 ADDR and CTL Fields for Normal Response Mode (Modulo 8)

Field	SDLC	Bit	Information Carried
ADDRESS		0-7	The address of the Gateway-ST for TRANSMIT COMPLETE and RECEIVE COMPLETE. This is the SDLC tributary address for the specified circuit. This address matches the address defined for the specified circuit when you configure OpenVMS for Gateway operations.
CTL	Information	0-2	Sequence number of message expected to be received.
		3	If set ON when a message is received by the Gateway system, it is a poll bit (primary station is polling). If set ON when a message is sent by the Gateway-ST, it is a final bit (secondary station completes response with this transmission).
		4-6	Sequence number of message sent.
		7	0 This is an information frame.
	Supervisory (indicates availability of station)	0-2	Sequence number of message expected to be received.
		3	If set ON when a message is received by the Gateway system, it is a poll bit (primary station is polling). If set ON when a message is sent by the Gateway-ST, it is a final bit (secondary station completes response with this transmission).
		4-5	Code for supervisory command or response. Codes and command/responses are: 00 = Receive ready 01 = Receive not ready 10 = Reject
		6-7	01 This is a supervisory frame.
	Unnumbered (manages data link)	0-2	Codes for unnumbered commands or responses (with bits 4-5) are listed and defined at the end of this table.
		3	If bit 3 is set ON when a message is received by the Gateway-ST, the primary station is polling.

(continued on next page)

Table 12–1 (Cont.) ADDR and CTL Fields for Normal Response Mode (Modulo 8)

Field	SDLC	Bit	Information Carried
			If bit 3 is set ON when a message is sent by the Gateway system, the secondary station completes the response with this transmission.
		4-5	Codes for unnumbered command or response are listed at the end of this table.
		6-7	11 (This is an unnumbered format frame.)
			Command Codes: 001P0011 = Unnumbered poll 000P0011 = Unnumbered information 100P0011 = Set normal response mode 010P0011 = Disconnect 101P1111 = Exchange station identification 111P0011 = Test 110P1111 = Set normal response mode extended
			Response Codes: 000F0011 = Unnumbered information 011F0011 = Unnumbered acknowledgment 000F1111 = Disconnect mode 100F0111 = Frame reject 101F1111 = Exchange station identification 111F0011 = Test
			where P represents either a 1 or a 0 poll bit and F represents a 1 or a 0 final bit.

Table 12–2 ADDR and CTL Fields for Extended Reponse Mode (Modulo 128)

Field	SDLC	Bit	Information Carried
ADDRESS		0-7	The address of the Gateway-ST for TRANSMIT COMPLETE and RECEIVE COMPLETE. This is the SDLC tributary address for the specified circuit. This address matches the address defined for the specified circuit when you configure OpenVMS for Gateway operations.
CTL	Information	0-6	Sequence number of message sent.
		7	0 This is an information frame.
		8-14	Sequence number of message expected to be received.
		15	If set ON when a message is received by the Gateway system, it is a poll bit (primary station is polling). If set ON when a message is sent by the Gateway-ST, it is a final bit (secondary station completes response with this transmission).
	Supervisory (indicates availability of station)	0-3	0000 always
		4-5	Code for supervisory command or response. Codes and command/responses are: 00 = Receive ready 01 = Receive not ready 10 = Reject
		6-7	01 This is a supervisory frame.
		8-14	Sequence number of message expected to be received.
		15	If set ON when a message is received by the Gateway system, it is a poll bit (primary station is polling). If set ON when a message is sent by the Gateway-ST, it is a final bit (secondary station completes response with this transmission).

(continued on next page)

Table 12–2 (Cont.) ADDR and CTL Fields for Extended Reponse Mode (Modulo 128)

Field	SDLC	Bit	Information Carried
	Unnumbered (manages data link)	0-2	Codes for unnumbered commands or responses (with bits 4-5) are listed and defined at the end of this table.
		3	If bit 3 is set ON when a message is received by the Gateway-ST, the primary station is polling.
			If bit 3 is set ON when a message is sent by the Gateway system, the secondary station completes the response with this transmission.
		4-5	Codes for unnumbered command or response are listed at the end of this table.
		6-7	11 (This is an unnumbered format frame.)
			Command Codes: 001P0011 = Unnumbered poll 000P0011 = Unnumbered information 100P0011 = Set normal response mode 010P0011 = Disconnect 101P1111 = Exchange station identification 111P0011 = Test 110P1111 = Set normal response mode extended
			Response Codes: 000F0011 = Unnumbered information 011F0011 = Unnumbered acknowledgment 000F1111 = Disconnect mode 100F0111 = Frame reject 101F1111 = Exchange station identification 111F0011 = Test
			where P represents either a 1 or a 0 poll bit and F represents a 1 or a 0 final bit.

12.4.3 Channel-Attached Circuit Level Trace

A channel-attached circuit level trace returns values for a command field (CMD) and a status field (STS). CMDs are always transmitted from the channel to the Gateway-CT. STSs are always transmitted from the Gateway-CT to the channel. See Table 12–3 and Table 12–4 for information about these values.

Each channel-initiated command sequence produces two events (R and T) in the trace data. An 'R' line is an event in which the command is presented to the Gateway-CT. For some write-type commands, data transfer may occur at this time. A 'T' line is an event in which the Gateway-CT responds to the command. For some read-type commands, data transfer may also occur at this time. Status may be sent to the channel when either or both of these events occur.

Each SNATRACE message in Example 12–2 is marked by a callout number. A corresponding number and explanatory text for that item follow the example.

Note

The sample SNATRACE messages in Example 12–2 show only the messages returned for a channel-attached circuit level trace.

Example 12–2 SNATRACE Output for Channel

```
$  RUN SYS$SYSTEM:SNATRACE

SNATRACE> TRACE/ANALYZE NODEG"SYSTEM SECRET":: CHAN-1/CIRCUIT

  SNATRACE Version 2.0    Circuit Trace    18-JAN-1988 14:37:51.93
  Gateway NODEG   Circuit CHAN-1  Type CHANNEL
  (Protocol Version = 2.0.0, Buffering level = 10, Data size = 256)

1    T 19:17:38.64                   Cmd (00) Sts (80)   (005,00000001)
2    R 19:17:38.64                   Cmd (32) Sts (08)   (005,00000001)
3    T 19:17:38.65                   Cmd (32) Sts (04)   (005,00000001)
4    R 19:17:38.66                   Cmd (02) Sts (00)   (005,00000001)
5    T 19:17:38.66 Count=36. bytes   Cmd (02) Sts (8D)   (005,00000001)
                TH=2C0000030001 RH=0B8000   RU=23. bytes
                FID2,OS,DAF=00,OAF=03,SNF=0001
                RQ,FMD,FI,BCI,ECI,DR1I
                0106 8100 E7E3 D3C1 D9C7 C540 F306 C3C9 : .\a.XTLARGE 3\CI
                C3E2 F1F7 0000 00                       : CS17 ...

6    R 19:17:38.66                   Cmd (52) Sts (08)   (005,00000001)
7    T 19:17:38.67                   Cmd (52) Sts (05)   (005,00000001)
8    R 19:17:38.67                   Cmd (31) Sts (08)   (005,00000001)
9    T 19:17:38.68                   Cmd (31) Sts (04)   (005,00000001)
10   R 19:17:38.68 Count=16. bytes   Cmd (09) Sts (08)   (005,00000001)
                TH=2C0000030001 RH=0B8000   RU=3. bytes
                FID2,OS,DAF=03,OAF=00,SNF=0001
                +RSP,FMD,FI,BCI,ECI,DR1I
                0106 81                                   .\a
11   T 19:17:38.69                   Cmd (09) Sts (04)   (005,00000001)
```

1 The value T indicates that the message has been transmitted by the Gateway-CT. The CMD field shows a value of 00 because this message is not being transmitted in response to a received command, so the CMD field is not applicable. The STS field shows a value of 80 which indicates that the Attention bit is set in this presentation of asynchronous status.

The hexadecimal values in parentheses that constitute the final field in this message are function and status values. The first number is octal and is an internal function code. The second is hexadecimal and is an OpenVMS status code.

2 The second message has a value of R which indicates that the message has been received by the Gateway-CT. The CMD field shows a value of 32 indicating the IBM system is executing a READ-START 0 channel program. The STS field shows a value of 08 (Channel End).

3 The third message is in the same format as message 1. The STS field shows a value of 04 (Device End) to the channel for the READ-START 0 command received. Note that the CMD field means that this message is transmitted in response to a READ-START 0, not that a READ-START 0 command is transmitted.

4 The fourth message has the same format as message 2. The CMD field shows a value of 02 indicating that the channel is executing a READ command.

5 The fifth message is in the same format as message 1. This message contains 6 lines. The Count field indicates that the link header data size is 36 bytes. The CMD field, 02, shows that this data has been transmitted to the channel to complete a READ command. The STS field shows a value of 8D (Attention, Channel End, Device End, and Unit Exception).

The next line contains the TH/RH/RU element. The TH field is six bytes in length and contains the hexadecimal value 2C0000030001. The RH field is three bytes long and contains the hexadecimal value 0B8000. The RU field is variable in length. In this example, the RU field contains 23 (decimal) bytes of data.

Note

An asterisk (*) immediately following the word bytes in the RU field length indicates the complete message was not displayed. For example, if you specify a /SIZE parameter of 64 bytes, and 256 bytes are actually transmitted, only 64 bytes of data are displayed.

On the third line of the message, TH data is displayed beginning with the term FID2. This information is explained in the IBM documentation.

The fourth line shows the RH data, beginning with the term RQ. This information is also explained in the IBM documentation.

The final two lines in the message, beginning with the value 0106, represent the RU data. The RU data is printed before the colon in hexadecimal values and after the colon in EBCDIC values.

Note

Some messages do not contain data in the TH, RH, or RU fields.

6 The sixth message contains the same information as the second message.

7 The seventh message contains the same information as the third message. The STS field shows a value of 05 (Device End, Unit Exception).

8 The eighth message has the same format as message 2. The CMD field shows a value of 31 indicating that the IBM system is executing a WRITE-START 0 channel program. The STS field shows value of 08 (Channel End) in response to this command.

9 The ninth message has the same format as message 1. The STS field shows a value of 04 (Device End) to the channel for the WRITE-START 0 command it received.

10 The tenth message has the same format as message 2. This message contains 5 lines. The Count field indicates the link header data size is 16 (decimal) bytes. The CMD field shows a value of 09 indicating that data has been received from the channel in a WRITE-BREAK command.

11 The last message has the same format and information as message 9 and shows ending status sent in response to the WRITE-BREAK command.

12.4.4 STS and CMD Fields in Messages

This section contains explanations of the bits returned in the CMD and STS fields of circuit level trace messages for channel-attached circuits.

Table 12–3 lists the CMD field values and the channel commands associated with each value.

Table 12–3 CMD Field Values and Associated Commands

Value	Channel Command
01	Write
02	Read
04	Sense
05	Control
09	Write Break
31	Write Start 0
32	Read Start 0
51	Write Start 1
52	Read Start 1
93	Restart Reset
E4	Sense ID

Table 12–4 lists the STS field values and their associated conditions. The actual STS value shown in the SNATRACE listing may be the sum of more than one of the status values shown in the table.

Table 12–4 STS Field Values and Associated Conditions

Value	Name	Condition
80	Attention (A)	Indicates an inbound message has been readied for transmission to the host. The host responds by issuing a Read CCW sequence.
40	Status Modifier (SM)	Indicates to the host that the control unit is ready to receive data from the host or set in response to WRITE-BREAK command, as a request for a Read. Also set with Busy (see below) when control unit is busy.
20	Control Unit End (CUE)	Is set following a busy condition, after pending status is cleared or when the control unit is no longer busy, to indicate that the control unit is now free to accept a new command.

(continued on next page)

Table 12–4 (Cont.) STS Field Values and Associated Conditions

Value	Name	Condition
10	Busy (B)	Is set in initial status byte with the Status Modifier (SM) when the addressed control unit is busy. The control unit uses this sequence when it cannot respond to the normal channel initiated selection sequence. See CUE for the reset of the busy state.
08	Channel End (CE)	Indicates channel data transfer operations are completed. No error unless Unit Check (UC) is included.
04	Device End (DE)	Indicates that the control unit is ready to receive a new command.
02	Unit Check (UC)	Is set when an invalid program or equipment condition is detected by the control unit or the device. The host always responds to Unit Check status by issuing a Sense command for further definition of condition.
01	Unit Exception (UE)	Indicates that no data is available for a successive read.

13

SNANCP Commands

This chapter describes SNANCP commands used with the Gateway. The command descriptions give complete syntax and usage information for each command. Commands are in alphabetical order.

The SNANCP commands apply to both the DECnet SNA Gateway-ST and the DECnet SNA Gateway-CT. Any restrictions on use are noted in the Command Parameter section for a particular command.

CLEAR ACCESS NAME

The CLEAR ACCESS NAME command removes the specified parameters from the access name definition. CLEAR ACCESS NAME ALL removes the access name from the database. There are no default access names, and no default access-name parameters.

Format

CLEAR access-name-component parameter [...]

Access Name Components

ACCESS NAME *access-name-id*
Removes the specified access name or parameters for the access name. The access-name-id is a string of 1 to 16 alphanumeric characters; for example, NCCF.

KNOWN ACCESS NAMES
Removes all known access names or parameters for all known access names.

Command Parameters

ALL
Removes the definition of the specified access name(s). Do not use ALL with the other parameters.

APPLICATION
Removes the application name from the access name definition.

DATA
Removes the data parameter from the access name definition.

LOGON MODE
Removes the name of the logon mode table entry from the access name definition.

NOTE
Removes the note text from the access name definition.

PU
Removes all the PU/LU-list pairs from the access name definition.

Examples

1. `SNANCP> CLEAR ACCESS NAME TSO PU`

   This command removes all the PU/LU-list pairs that have been defined for the access name TSO.

2. `SNANCP> CLEAR ACCESS NAME TSO ALL`

   This command removes the access name TSO.

CLEAR CIRCUIT

The CLEAR CIRCUIT command resets circuit parameters to the default values (if any). CLEAR *circuit-component* ALL removes the definition of the specified circuit(s) from the database.

Before clearing a circuit or circuit parameters (except for COUNTER TIMER, LOGGING, and NOTE), make sure the circuit is in the OFF state.

Format

CLEAR   circuit-component parameter [...]

Circuit Components

CIRCUIT *circuit-id*

Resets parameters to the default values (if any) or removes the parameters for a specific circuit.

- When using SDLC circuits, enter the *circuit-id* in the form SDLC-*n*; for example, SDLC-0.
- When using channel-attached circuits, enter the *circuit-id* in the form CHAN-*n*; for example, CHAN-1.

KNOWN CIRCUITS

Resets parameters to the default values (if any) or removes the parameters for all known circuits.

Command Parameters

ALL

Removes the definition of the specified circuit or all known circuits. The circuit must be in the OFF state before you specify this parameter. Do not use ALL with the other parameters.

CHANNEL ADDRESS

Applies only to channel-attached circuits. Removes the channel address parameter. You must specify another channel address before you can turn the circuit to the ON state.

COUNTER TIMER

Resets the logging timer to the default value of 0 (this means that the timer is not running). This cancellation of the logging timer prevents any further circuit counter logging for the indicated circuit(s).

DUPLEX

Applies only to SDLC circuits. Resets the data transfer mode of the specified circuit to the default, which is HALF (half duplex).

The duplex setting must correspond with the DATMODE parameter for the IBM ACF/NCP PU macro.

For information on setting the physical characteristics of the communications path, see the SNANCP SET LINE command.

IDLE TIMER

Applies only to SDLC circuits. Resets the number of seconds the circuit should wait for a poll from the IBM system to the default value of 30.

LINE

Resets the line allocated to the circuit to the default value, which is none.

LOGGING

Resets logging to the default which is to log only fatal events. Fatal event messages are always logged, even when other levels are specified.

NOTE

Removes the note text for the specified circuit(s). The default is no note.

RESPONSE MODE

Applies only to SDLC circuits. Resets response mode to NORMAL. This setting must correspond to the ACF/NCP LINE macro MODULO parameter.

STATION ADDRESS

Applies only to SDLC circuits. Resets the SDLC station address to the default value of 01. This value must correspond with the setting of the ADDR parameter for the IBM ACF/NCP PU macro.

STATION ID

Applies only to SDLC circuits. Resets the SDLC exchange identification (XID) for dial-up lines to the default value of 00000000. The first three digits correspond with the setting of the IDBLK parameter for the IBM VTAM PU macro. The last five digits must correspond with the setting of the IDNUM parameter for the IBM VTAM PU macro.

Examples

1. `SNANCP> CLEAR CIRCUIT SDLC-0 ALL`

   The CLEAR CIRCUIT ALL command removes the definition of the circuit.

2. `SNANCP> CLEAR CIRCUIT SDLC-0 LOGGING`

   The CLEAR CIRCUIT LOGGING command resets logging to the default, which is to log only fatal events.

3. `SNANCP> CLEAR CIRCUIT CHAN-0 COUNTER TIMER`

   The CLEAR CIRCUIT COUNTER TIMER command resets the counter timer to the default of 0.

CLEAR LINE

The CLEAR LINE command resets line parameters to the default values (if any). CLEAR *line-component* ALL removes the definition of the specified line(s) from the database.

Before clearing a line or line parameters (except for COUNTER TIMER, LOGGING, and NOTE), make sure the line is in the OFF state.

Format

CLEAR line-component parameter [...]

Line Components

KNOWN LINES
Resets all known lines to default values (if any) or removes the parameters for all lines.

LINE *line-id*
Resets the specified line to default values (if any) or removes the parameters for a specific line. *line-id* specifies the name of the line you want to clear.

- When using SDLC lines, enter the *line-id* in the form SYN-*n*; for example, SYN-0.
- When using channel-attached lines, enter the *line-id* in the form CQ-*n*; for example, CQ-0.

Command Parameters

ALL
Removes the definition of the specified line(s). The line must be in the OFF state before you specify this parameter. Also, there must be no circuits associated with this line.

ATTENTION BUFFERS
Applies to channel-attached lines only. Resets the number of attention buffers to the default value, 12.

BUFFER SIZE
Resets the size of the buffers that receive messages from the IBM system to the default value.

- When using SDLC lines, the default value is 265.

- When using channel-attached lines, the default value is 4105.

CLOCK
Applies only to SDLC lines. Resets the clocking that regulates signals between the IBM system and the Gateway-ST to the default value, which is EXTERNAL.

COUNTER TIMER
Resets the logging timer to the default value of 0 (this means that the timer is not running). This cancellation of the logging timer prevents any further line counter logging for the indicated line(s).

DUPLEX
Applies only to SDLC lines. Resets the duplex setting of the line to the default value, which is HALF (half duplex). The duplex setting must be the same as that defined for the IBM modem by the value of the DUPLEX parameter and by the ADDRESS parameter for the IBM ACF/NCP LINE macro.

FORCED BURST
Applies only to channel-attached lines. Resets forced burst mode to the default of DISABLED.

INTERLEAVE
Applies only to channel-attached lines. Resets a byte multiplexer channel's interleave size to the default value of 16 bytes. This parameter applies only to channels using the channel byte protocol with the FORCED BURST parameter DISABLED.

LOGGING
Resets LOGGING to the default value, which is to log only fatal events. Fatal event messages are always logged, even when other levels are specified.

MODEM TYPE
Applies only to SDLC lines. Resets the modem type to the default value of NORMAL (for transfer of data).

NOTE
Removes the note text for the specified line(s). The default is no note.

PROTOCOL
Resets the protocol to the default value. For SYN-*n* lines, the default is SDLC POINT. For CQ-*n* lines, the default is CHANNEL BLOCK.

RECEIVE BUFFERS

Resets the maximum number of buffers to be allocated to the line to the default value.

- When using SDLC lines, the default value is 34.
- When using channel-attached lines, the default value is 64.

SIGNALLING

Resets the mode for line signals to the default value of NORMAL. When using this parameter with SDLC lines, the value must correspond with the setting of the NRZI parameter for the IBM ACF/NCP LINE macro for synchronous lines. When using this parameter with channel-attached lines, the values must match the capabilities of the channel hardware.

Examples

1.
```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> CLEAR LINE SYN-0 ALL
```

After setting line SYN-0 to the OFF state, the CLEAR LINE ALL command removes the definition of the line.

2.
```
SNANCP> CLEAR LINE SYN-0 NOTE
```

The CLEAR LINE NOTE command removes the note text for the line.

3.
```
SNANCP> SET LINE CQ-0 STATE OFF
SNANCP> CLEAR LINE CQ-0 FORCED BURST
```

After setting line CQ-0 to the OFF state, the CLEAR LINE FORCED BURST command resets forced burst mode to the default of DISABLED.

CLEAR LOGGING

The CLEAR LOGGING command removes or resets parameters to default values for all or some event logging. Set the logging state to OFF before using the CLEAR LOGGING command.

See the CLEAR command for individual components (LINE, CIRCUIT, PU, LU, ACCESS NAME, SERVER) for a description of how to reset the level of messages logged for a particular component.

Format

CLEAR   logging-component parameter [...]

Logging Components

KNOWN LOGGING
Removes or resets parameters for all known logging components.

LOGGING CONSOLE
Removes parameters for logging to a dedicated console or text file.

LOGGING FILE
Removes parameters for logging to a binary file.

LOGGING MONITOR
Removes or resets parameters for logging to a monitor.

Command Parameters

ALL
Removes parameters for all known logging components.

NAME
For the LOGGING CONSOLE component, removes the name of the console or text file to which messages are sent. There is no default.

For the LOGGING FILE component, removes the name of the file to which messages are logged. There is no default file.

For the LOGGING MONITOR component, removes the name of the monitor to which messages are sent. Messages are only sent to OPCOM.

SINK NODE *node-id*
Specifies the DECnet node for which event logging parameters are cleared. If SINK NODE is not specified, the default is the load host node.

KNOWN SINKS
Specifies that the logging parameters are to be cleared for all sink nodes.

Example

```
SNANCP> CLEAR LOGGING MONITOR SINK NODE NODEG NAME
```

This command removes the name of the monitor that logs the event messages. Event messages are sent only to the default monitor, OPCOM.

CLEAR LU

The CLEAR LU command resets LU parameters to the default values (if any). CLEAR LU ALL resets all LU parameters to their default values. (To remove the definition of an LU, use the CLEAR PU LU LIST command.)

If the LU is being used when you clear or change the authorization entry, the current session is not affected. The Gateway checks for the new information the next time it attempts to allocate the LU.

Format

CLEAR lu-component [component-qualifier] parameter [...]

LU Components

KNOWN LUS
Resets parameters for all known LUs.

LU *lu-id*
Resets parameters of the specified LU to default values. Enter the *lu-id* in the form *pu-id.nnn*; for example, SNA-0.1.

Component Qualifiers

AUTHORIZATION *auth-id*
Removes the definition for a specific authorization entry or all known authorization entries.

KNOWN AUTHORIZATIONS
Removes the definitions for all known authorization entries.

You can remove any or all of the following access control parameters for an authorization entry (see the SNANCP SET LU command for a description of these parameters):

NODE	Removes the name of the DECnet node from which the user is trying to access the LU.
USER	Removes the name of the remote user trying to access the LU.
TERMINAL	Removes the name of the terminal from which a user is trying to access the LU.

PASSWORD	Removes the password assigned to the authorized user for this LU.
ALL	Removes the entire authorization entry.

If you clear one of the CLEAR LU parameters, that particular parameter is not checked before access to the Gateway is accepted.

There is no authorization information by default.

You must enter the CLEAR LU AUTHORIZATION component qualifier as a separate command. It cannot be part of a command that clears LU parameters.

Command Parameters

ALL
Resets all parameters to their default values.

LOGGING
Resets LOGGING to the default, which is to log only fatal events. Fatal event messages are always logged even when other levels are specified.

NOTE
Removes the note text for the specified LU(s). The default is no note.

Examples

1.
```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> CLEAR LU SNA-0.1
```

   The first command sets the circuit SDLC-0 (the circuit associated with PU SNA-0) to the OFF state. The second command resets logging to the default value (fatal) and removes any note text for LU SNA-0.1.

2.
```
SNANCP> CLEAR LU SNA-0.1 AUTHORIZATION NEW_AUTH NODE USER
```

   This command removes the NODE and USER information from the authorization entry NEW_AUTH.

3.
```
SNANCP> CLEAR LU SNA-0.1 AUTHORIZATION NEW_AUTH ALL
```

   This command removes the entire authorization entry NEW_AUTH from the database.

CLEAR PU

The CLEAR PU command resets PU parameters to the default values (if any). CLEAR *pu-component* ALL removes the definition of the specified PU(s) from the database.

Before clearing a PU or PU parameters, make sure the corresponding circuit is in the OFF state.

Format

CLEAR pu-component parameter [...]

PU Components

KNOWN PUS
Resets all PUs to default values (if any) or removes the definitions for all PUs.

PU *pu-id*
Resets the specified PU to default values (if any) or removes the definition for a specific PU. Enter the *pu-id* in the format *SNA-nnn*; for example, SNA-0.

Command Parameters

ALL
Removes the definition of the specified PU or all known PUs (including any associated LUs). A PU must be inactive before you remove its definition or delete an LU list. To make a PU inactive, set the state of the corresponding circuit to OFF.

CIRCUIT
Deallocates the circuit for this PU. You can no longer use the cleared circuit for the specified PU.

COUNTER TIMER
Resets the logging timer to the default value of 0 (this means that the timer is not running). This cancellation of the logging timer prevents any further PU counter logging for the specified PU(s).

LOGGING
Resets LOGGING to the default, which is to log only fatal events. Fatal event messages are always logged, even when other levels are specified.

LU [LIST]
Removes the definition of all LUs associated with this PU.

NOTE
Removes the note text for the PU. The default is no note.

[QUERY] THRESHOLD
Resets the number of seconds that must elapse between REQMS (Request Maintenance Statistics) messages to the default value of 0. When the value is 0, the Gateway processes all REQMS messages it receives from the IBM Network Problem Determination Application (NPDA).

See Appendix B for information on REQMS messages.

SEGMENT [SIZE]
Resets the size of the frames of data sent to the IBM system to the default value of 265. This parameter is not used with PUs associated with channel-attached circuits.

SSCP [ID]
Clears the SSCP list that had been defined for this PU.

Example

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> CLEAR PU SNA-0 LU LIST SSCP ID
```

After you set the state of circuit SDLC-0 to OFF, the CLEAR PU command removes all SSCP IDs and the LU list defined for PU SNA-0. You must use the SET command to define a new LU LIST and new SSCP IDs.

CLEAR SERVER

The CLEAR SERVER command resets parameters to their default values. The CLEAR *server-component* ALL command does not delete a server; rather, it resets all parameters to their default values.

For all servers, set the state to OFF before you reset the parameter values with the CLEAR SERVER ALL command. For the DHCF server, set the server state to OFF before you clear the MAXIMUM LINKS and PU parameters.

Format

CLEAR server-component [component-qualifier] parameter [...]

Server Components

KNOWN SERVERS

Resets parameters for all known servers.

SERVER *server-id*

Resets the specified parameters for a particular server. The server-id is one of the following:

- RJE
- SNA-ACCESS
- DHCF

Component Qualifiers

KNOWN LUS

Removes the association of all KNOWN LUS for outbound session initiations.

LU *lu-id*

Removes the association of the specified LU for outbound session initiations.

You can remove any of the following outbound connection initiation parameters from the LU(s) allocated to the SNA-ACCESS server:

NODE	Removes the name of the DECnet node that is to receive the BIND request.
OBJECT NAME	Removes the string that identifies the DECnet object that is to receive the BIND request.

OBJECT NUMBER	Removes the number that identifies the DECnet object that is to receive the BIND request.

You can also remove the following parameters that define access control information for establishing a logical link to the DECnet node:

USER	Removes the user identification that is used for establishing the logical link.
PASSWORD	Removes the password that is used to establish the logical link.
ACCOUNT	Removes the user account that is used for establishing the logical link.

You can remove the entire outbound server allocation information by specifying ALL.

Command Parameters

ALL
Removes the specified server(s) from the database.

LOGGING
Resets the LOGGING parameter to the default, which is to log only fatal event messages. Fatal event messages are always logged, even when other levels are specified.

MAXIMUM LINKS
Resets the maximum number of DECnet logical links that can be active for the server. The maximum value that is set is server and system dependent.

NOTE
Removes note text from the server definition. The default is no note.

The following parameters apply only to the DHCF server:

PU pu-id LU [LIST]
Removes the specified PU and associated LU list from the set of LUs allocated to the DHCF server. Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0.

TEXT FILE *file-spec*
Removes the current text file. You can use the SET SERVER *server-id* TEXT FILE command to define a new file.

Examples

1. ```
   SNANCP> SET SERVER DHCF STATE OFF
   SNANCP> CLEAR SERVER DHCF PU SNA-0 LU LIST
   ```

   The first command sets the server state to OFF. The second command deallocates all LUs on PU SNA-0 for the DHCF server.

2. ```
   SNANCP> CLEAR SERVER SNA-ACCESS LU SNA-0.1 ALL
   ```

   This command removes all outbound connection parameters from LU SNA-0.1.

EXIT

The EXIT command terminates the SNANCP image and returns you to DCL level.

You can leave SNANCP by entering the EXIT command and pressing the Return key or by pressing Ctrl/Z.

Example

```
SNANCP> EXIT Return
or
SNANCP> Ctrl/Z
```

Both of these commands terminate the SNANCP image and return you to DCL level.

HELP

The HELP command provides information on SNANCP commands. If you enter just the word HELP, you see a display of information available within SNANCP. If you enter a command name after the word HELP, you get information on the specified command.

Example

```
SNANCP> HELP SET CIRCUIT
```

This command provides information on the SET CIRCUIT command.

LOOP LINE

The LOOP LINE command tests the communications hardware and software. *The LOOP LINE command and its parameters apply only to SDLC lines.* See *DECnet SNA Gateway-ST Problem Solving* for more information on loopback tests on SDLC lines.

Format

LOOP   line-component parameter [...]

Line Component

LINE *line-id*
Specifies the line that is to be tested. Enter the *line-id* in the form SYN-*n*; for example, SYN-0.

Command Parameters

AT CONTROLLER
Applies only to SDLC lines. Tests the communications device. If this test fails, there is a fault with the communications device.

If this test is successful, use the AT CABLE parameter to try to isolate the problem.

AT CABLE
Applies only to SDLC lines. Tests the cable from the Gateway-ST to the cable loopback connector. If this test fails, there is a fault with the cable. Replace the cable.

If this test is successful, use the AT MODEM parameter to try to isolate the problem.

AT MODEM
Applies only to SDLC lines. Tests the local or remote modem. Set the modem you want to test to analog loopback mode. Then enter the LOOP LINE *line-id* AT MODEM command. If this test fails, repair or replace the modem.

If this test is successful, but you still have a problem, you might have to arrange for tests to be carried out from the IBM side. Speak to your IBM systems programmer to arrange for such tests.

LOOP LINE

Example

```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> LOOP LINE SYN-0 AT CONTROLLER
```

After you set line SYN-0 to the OFF state, the LOOP LINE command tests the communications device.

SET ACCESS NAME

The SET ACCESS NAME command lets you create a name that represents a list of parameters that are needed to access an IBM SNA system. There are no default access names and no default parameters for access names.

Format

SET access-name-component parameter [...]

Access Name Components

ACCESS NAME *access-name-id*
The access-name-id is a string of 1 to 16 alphanumeric characters; for example, NCCF.

KNOWN ACCESS NAMES
Defines or changes parameters for all known access names.

Command Parameters

The following table lists the parameters and value ranges for access name parameters:

Parameter	Value	Default
APPLICATION	8 chars	None
DATA	32 chars	None
LOGON MODE	8 chars	None
LU [LIST]	List, each 1-255	1-255
NOTE	30 chars	None
PU	SNA-*nnn*	None

APPLICATION *ibm-applid*
Specifies (in no more than 8 characters) the name of the IBM application to be accessed. Ask the IBM systems programmer for the name of the application.

DATA *logon-data*
Specifies (in no more than 32 characters) user-specific information for a session. If the data text contains spaces, enclose the data in quotation marks. Not all IBM applications require the data specified with this parameter. If you need

to specify data for this parameter, ask the IBM system programmer what it should be.

For security reasons, the data associated with this parameter is not shown when nonprivileged users enter a SHOW ACCESS NAME command. To have this information displayed on the screen, you must specify a privileged username and password with the USE SYSTEM command.

Nevertheless, the access name definition can be still used by unprivileged users. Keep this in mind when including sensitive information in access names.

LOGON MODE *name*
Specifies (in no more than 8 characters) the name of a logon mode table entry. The logon mode table entry is one of several entries in a logon mode table. Each entry is a set of rules and protocols. These entries are used by IBM applications to define how a session is to be conducted. The default table entry could be adequate for your purposes. Ask the IBM systems programmer for the logon mode table entry.

LU [LIST] *lu-list*
Specifies a list of LUs that are used to access the IBM system. You must specify the associated PU for an LU list. The LU list should contain LU numbers or ranges of LU numbers separated by commas. An example of an LU LIST is 1–10,54,15–34. The LU LIST you specify replaces any list of LUs already defined for the associated PU.

You can specify the LUs for only one PU at a time. To specify LU lists for more than one PU, issue the command a separate time for each PU. With the SET ACCESS NAME command, the LU numbers that you list must be within the range of valid LU numbers defined by the SET PU *pu-id* LU LIST command. The order in which LUs appear in the LU LIST has no bearing on the order in which they are used (the use-order is undefined).

NOTE *string*
Lets you specify additional information (in no more than 30 characters) that will appear in the SHOW SUMMARY display. For example, you can add information about the IBM application to which the access name refers. Specify the text of your note within quotation marks.

PU *pu-id*
Specifies the PU that is used to access the IBM system. Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0. At least one PU must be specified per access name. You can specify only one PU in a single command; issue a separate command for each PU that you want to include in the access name.

Example

```
SNANCP> SET ACCESS NAME CICS PU SNA-0 LU LIST 1-64 -
_SNANCP> APPLICATION CICS NOTE "access to CICS"
```

This command defines an access name for CICS by specifying values for the PU, LU LIST, APPLICATION, and NOTE parameters.

```
SNANPC> SET ACCESS NAME NETVIEW APPLICATION NCCF2
SNANCP> SET ACCESS NAME NETVIEW PU SNA-0 LU LIST 1-10
SNANCP> SET ACCESS NAME NETVIEW PU SNA-1 LU LIST 5-15
```

The above sequence of commands defines an access name NETVIEW, which allows two different PUs to be used for the connection. A different LU list is associated with each PU.

SET CIRCUIT

The SET CIRCUIT command defines or changes circuits and their parameters. Be sure that the circuit state is OFF before using SET CIRCUIT (except for COUNTER TIMER, NOTE, and LOGGING).

Format

SET circuit-component parameter [...]

Circuit Components

CIRCUIT *circuit-id*
Defines or changes parameters for the circuit you specify.

- When using SDLC circuits, enter the circuit-id in the form SDLC-*n*; for example, SDLC-0.
- When using channel-attached circuits, enter the circuit-id in the form CHAN-*n*; for example, CHAN-1.

KNOWN CIRCUITS
Defines or changes parameters for all known circuits.

Command Parameters

The following are default values for circuit parameters [1]:

Parameter	Value	Default
CHANNEL ADDRESS	3 hexadecimal digits	None
COUNTER TIMER	0-65535	0
DUPLEX	HALF or FULL	HALF
IDLE TIMER	15-1800	30
LINE	SYN-*n*	None
	CQ-*n*	None
LOGGING	FATAL,ERROR	FATAL

[1] Any restrictions on use are noted in the command parameter definitions.

Parameter	Value	Default
	WARNING, or INFORMATIONAL	
NOTE	0-30 chars	No note
RESPONSE MODE	NORMAL or EXTENDED	NORMAL
STATE	ON or OFF	OFF
STATION ADDRESS	2 hex digits	01
STATION ID	8 hex digits	00000000

CHANNEL ADDRESS *3 hex digits*
Applies only to channel-attached circuits. Specifies the channel address for the Gateway-CT. Obtain the channel address from your IBM systems programmer. This parameter is specified as three hexadecimal digits. There is no default channel address. You cannot set the circuit state to ON before you have specified a channel address.

COUNTER TIMER *seconds*
Specifies how long (0–65535 seconds) the counter timer should run. If event logging is enabled for information level messages, when the timer expires, the counters are logged to the DECnet host and the counters are set to zero.

The default is 0, which means that the timer is not running.

DUPLEX *keyword*
Applies only to SDLC circuits. Controls how data is transferred. Specify whether the data transfer is to be two-way simultaneous (FULL) or two-way alternate (HALF). On a circuit in full-duplex mode, frames can be sent and received at the same time. On a half-duplex circuit, each end must wait until the other end has finished sending before it can send.

The circuit duplex must agree with the DATMODE parameter in the IBM ACF/NCP GROUP or PU macro. If the DATMODE parameter is specified as FULL, the circuit is full-duplex and if the DATMODE parameter is specified as HALF, the circuit is half-duplex.

The following combinations of circuit duplex and line duplex are allowed:

- A full-duplex line and a half-duplex circuit
- A full-duplex line and a full-duplex circuit
- A half-duplex line and a half-duplex circuit

Remember, for full-duplex data transfer, not only the circuit but also the line must be operating in full-duplex mode. If you set the circuit to FULL, the corresponding line duplex parameter must also be set to FULL. See SET LINE command. However, you can run a half-duplex circuit over a full-duplex line if you wish.

The default for circuit DUPLEX is HALF.

IDLE TIMER *seconds*
Applies only to SDLC circuits. Specifies the number of seconds (15–1800) the circuit should wait for a poll from the IBM system. If the idle timer expires, the circuit goes to the ON–TIMED OUT state, and, if logging is enabled, an event message is logged to the DECnet host.

The default is 30 seconds.

LINE *line-id*
Allocates a line to the circuit. You cannot set a circuit ON until you allocated a line for the circuit. You can also use this parameter to switch the same circuit between different lines.

- When using SDLC lines, enter *line-id* in the form SYN-*n*; for example, SYN-0. Each SDLC line supports a single circuit.
- When using channel-attached lines, enter *line-id* in the form CQ-*n*; for example, CQ-0. Each channel-attached line supports several circuits.

LOGGING [ENABLED] *keyword*
Specifies which levels of messages are recorded. The keyword is one of the following:

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

The level you specify sets the minimum level to be logged; all levels of greater severity will also be logged. For example, if you set logging to INFORMATIONAL, messages at all other levels will also be logged. Fatal event messages are always logged. For more information on event messages, severity levels, and logging commands, see Chapter 10.

NOTE *string*
Lets you specify additional information (in no more than 30 characters) that will be part of the SHOW SUMMARY display. Specify the text of your note within quotation marks.

The default is no note.

RESPONSE MODE *keyword*
Applies only to SDLC circuits. Specifies the frame numbering of the circuit as one of the following:

- NORMAL (modulo 8)
- EXTENDED (modulo 128)

Modulo 128 frame numbering is supported only on IBM ACF/NCP Version 4, and the setting must correspond to the ACF/NCP LINE macro MODULO parameter.

The default is NORMAL.

STATE *keyword*
Specifies the operational state of the circuit as ON or OFF. The circuit must be ON for Gateway operations.

The default is OFF.

STATION ADDRESS *2 hex digits*
Applies only to SDLC circuits. Specifies the SDLC station (or tributary) address that the IBM system uses to address the station on the line. Enter this address as 2 hexadecimal digits. The value must correspond with the setting of the ADDR parameter for the IBM ACF/NCP PU macro.

The default is 01.

STATION ID *8 hex digits*
Applies only to SDLC circuits. This parameter is for dial-up lines only. This SDLC Exchange-Identity (XID) string for the circuit, is specified as 8 hexadecimal digits. It is returned to the IBM system when an XID command has been received. The first 3 digits must correspond with the setting of the IDBLK parameter for the IBM VTAM PU macro. The last 5 digits must correspond with the setting of the IDNUM parameter for the IBM VTAM PU macro.

The default is 00000000.

Examples

1.
```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> SET CIRCUIT SDLC-0 STATE ON DUPLEX HALF -
_SNANCP> LINE SYN-0 NOTE "IBM Line L022B"
```

The first command sets the SDLC-0 circuit to the OFF state. The second command modifies the values of STATE, DUPLEX, LINE, and NOTE.

2.
```
SNANCP> SET CIRCUIT CHAN-2 STATE OFF
SNANCP> SET CIRCUIT CHAN-2 STATE ON COUNTER TIMER -
_SNANCP> 785 LINE CQ-0 NOTE "Byte Multiplexer Channel link"
```

The first command sets the CHAN-2 circuit to the OFF state. The second command modifies the values of STATE, COUNTER TIMER, LINE, and NOTE.

3.
```
SNANCP> SET CIRCUIT CHAN-3 LOGGING WARNING
```

This command specifies that warning events and events of greater severity (error and fatal) are to be logged. Events of lesser severity (informational) are not logged.

SET LINE

The SET LINE command defines or changes lines and their parameters. SNANCP manages lines from the Gateway to the IBM system.

Make sure that the line state is OFF before using SET LINE to define line parameters (except for COUNTER TIMER, NOTE, and LOGGING).

Format

SET line-component parameter [...]

Line Components

KNOWN LINES
Defines or changes parameters for all known lines.

LINE *line-id*
Defines or changes parameters for the line you specify.

- When using SDLC lines, enter the *line-id* in the form SYN-*n*; for example, SYN-0.
- When using channel-attached lines, enter the *line-id* in the form CQ-*n*; for example, CQ-0.

Command Parameters

Table 13–1 lists the default values for line parameters. Any restrictions on use are noted in the command parameter definitions.

Table 13–1 Default Values for Server Parameters

Parameter	Value	Default
ATTENTION BUFFERS	2-128	12
BUFFER SIZE	*For SDLC lines*: 265-1482 bytes	265

(continued on next page)

Table 13–1 (Cont.) Default Values for Server Parameters

Parameter	Value	Default
	For Channels: 265-8250 bytes	4105
CLOCK	EXTERNAL or INTERNAL	EXTERNAL
COUNTER TIMER	0-65535	0
DUPLEX	HALF or FULL	HALF
FORCED BURST	DISABLED or ENABLED	DISABLED
INTERLEAVE	2-31 bytes	16 bytes
LOGGING	FATAL,ERROR WARNING, or INFORMATIONAL	FATAL
MODEM TYPE	NORMAL or DIAGNOSTIC	NORMAL
NOTE	0-30 chars	No note
PROTOCOL	*For SDLC lines*: SDLC POINT or SDLC TRIBUTARY	SDLC POINT
	For Channels: CHANNEL BYTE, CHANNEL BLOCK or CHANNEL SELECTOR	CHANNEL BLOCK
RECEIVE BUFFERS	16-512	*For SDLC lines*: 34
		For Channels: 64
SIGNALLING	*For SDLC lines*: NRZI or NORMAL	NORMAL
	For Channels: NORMAL or HIGH SPEED	NORMAL
STATE	ON or OFF	OFF

ATTENTION BUFFERS

Applies only to channel-attached lines. This parameter sets the maximum number of buffers allocated to the line driver for capturing certain events such as command receipt on the channel.

Digital recommends that you configure at least two attention buffers per circuit. For example, if you have two circuits on your channel-attached line, you should configure at least four attention buffers. The default value is 12.

You can check the "local buffer errors" and "attention buffers at peak" counters to see if you have too few attention buffers.

BUFFER SIZE *bytes*
Specifies the maximum size of the buffers used to receive frames transmitted by the IBM host.

When using SDLC lines, specify a value from 265 to 1482 bytes. The BUFFER SIZE value must match the value specified for MAXDATA in the IBM ACF /NCP PU macro. The PU SEGMENT SIZE parameter is related to BUFFER SIZE.

The default is 265 bytes.

When using channel-attached lines, specify a value from 265 to 8250 bytes. The specified size must exceed the maximum RU size by at least 9 bytes. Thus, if you want to use a maximum RU size of 2K bytes, specify a buffer size of 2057 bytes (2048 bytes plus an additional 9 bytes).

The default is 4105 bytes.

CLOCK *keyword*
Applies only to SDLC lines. Specifies whether clocking is INTERNAL or EXTERNAL. Clocking regulates the signals sent between the IBM system and the Gateway-ST.

INTERNAL mode causes the DEC MicroServer synchronous port to supply a clock signal. Use this mode of operation with loopback tests to allow transmitted messages to be looped back through a passive loopback connector. Use INTERNAL mode for the CLOCK parameter only when testing.

For normal data transfer, use EXTERNAL mode. When you specify EXTERNAL mode, the clock signal must be supplied externally to the synchronous port.

The default is EXTERNAL.

COUNTER TIMER *seconds*
Specifies how long (0–65535 seconds) the counter timer should run. If event logging is enabled for information messages, when the timer expires, the counters are logged to the DECnet host and the counters are set to zero.

The default is 0, which means the timer is not running.

DUPLEX *keyword*

Applies only to SDLC lines. Controls how modem signaling works. The duplex mode of the line can be HALF (half duplex) or FULL (full duplex). The duplex mode must be the same as that defined for the IBM modem. To determine the IBM modem setting, refer to the DUPLEX and ADDRESS parameters for the IBM ACF/NCP LINE macro.

The line DUPLEX parameter sets only the physical characteristics of the communications path. Setting the line to full duplex does not necessarily set the line protocol to full duplex. To set the data transfer mode to full or half duplex, see the SNANCP SET CIRCUIT command.

The default is HALF.

FORCED BURST *keyword*

Applies only to channel-attached lines. When using byte multiplexer channels, this parameter specifies whether the Gateway-CT should operate in forced burst mode. The keyword value is either ENABLED or DISABLED.

When forced burst mode is ENABLED, the DEC ChannelServer maintains ownership of channel resources for the transfer of an entire data block.

When forced burst mode is DISABLED, the DEC ChannelServer rearbitrates ownership every time a few bytes are transferred. The rearbitration of ownership allows fair access to the channel but it also lowers the total throughput rate for the Gateway-CT.

INTERLEAVE *bytes*

Applies only to channel-attached lines. The INTERLEAVE parameter is used only with byte multiplexer channels not operating in forced burst mode. This parameter specifies the number of data bytes transferred before the Gateway-CT relinquishes channel resources to other devices.

The interleave factor ranges from 2 to 31 bytes. The default is 16 bytes.

For byte multiplexer channels operating in forced burst mode, and for other IBM channel links, the INTERLEAVE parameter is accepted but ignored. The INTERLEAVE parameter does not apply to synchronous communication links.

LOGGING [ENABLED] *keyword*

Specifies which of levels of event messages are recorded. The keyword is one of the following:

- INFORMATIONAL
- WARNING
- ERROR

- FATAL

The logging level you specify with a SET LINE command replaces the logging level previously defined. Fatal event messages are always logged.

MODEM TYPE *keyword*
Applies only to SDLC lines. Specifies the type of modem as one of the following:

- NORMAL (without IBM diagnostic capability)
- DIAGNOSTIC (with IBM diagnostic capability)

IBM modems (such as the 386*x*) offer diagnostic capabilities to the host system. Other modems typically do not have such features.

This parameter must agree with both the type of modem and the setting of the LPDATS parameter for the IBM ACF/NCP LINE macro. If LPDATS=NO, set the modem type to NORMAL. If LPDATS=YES, set the modem type to DIAGNOSTIC.

Note

If you are not using diagnostic modems and you specify the modem type as DIAGNOSTIC, modems that use the test indicator (TI) lead as a clock degrade the performance of the Gateway-ST.

The default is NORMAL.

NOTE *string*
Lets you specify additional information (no more than 30 characters) that will be part of the SHOW SUMMARY display. Specify the text of your note within quotation marks.

PROTOCOL *keyword*
Defines the Data Link protocol to be used on this line. The following keywords can be used for the protocol name when using SDLC lines:

SDLC POINT	Defines this line as one end of a point-to-point SDLC connection.
SDLC TRIBUTARY	Defines this line as a tributary end of an SDLC multipoint group.

The default is SDLC POINT.

The following keywords can be used for the protocol name when using channel-attached lines:

CHANNEL BYTE	Designates a byte multiplexer channel.
CHANNEL BLOCK	Designates a block multiplexer channel.
CHANNEL SELECTOR	Designates a selector channel.

The default is CHANNEL BLOCK.

RECEIVE BUFFERS *number*

Specifies the maximum number of buffers (16-512) to be allocated to the line. These buffers are for holding messages from the IBM system.

When using SDLC lines, make the following calculations and use the larger of the two values as the parameter for RECEIVE BUFFERS:

- $(2 * MAXOUT) + 2$
- $2(Maximum\ RU\ size/(MAXDATA - 9)) + 2$

The default is 34.

The MAXOUT and MAXDATA values are specified in IBM ACF/NCP macros. Obtain these values from the IBM systems programmer.

The parameter value for RECEIVE BUFFERS is especially important when you are using circuits with SDLC extended response mode.

When using channel-attached lines, the number of buffers depends on the expected line activity. An insufficient number of buffers is indicated by an significant increase in the "local buffer errors" counter when there is activity on the line. The default is 64.

SIGNALLING *keyword*

Specifies the convention for line signals as one of the following:

- NORMAL
- NRZI
- HIGH SPEED

When using this parameter with SDLC lines, the signalling mode maintains bit synchronization. The parameter setting must correspond with the setting of the NRZI parameter for the IBM ACF/NCP LINE macro.

When using this parameter with channel-attached lines, the SIGNALLING parameter specifies the data transfer mode.

In NORMAL mode, data transfer uses only service-in and service-out tags.

In HIGH SPEED mode, data transfer uses data-in and data-out tags as well as service-in and service-out tags. You should only specify HIGH SPEED if your particular channel hardware supports this option.

STATE *keyword*
Specifies the operational state of the line as ON or OFF. The line must be ON for Gateway operations.

Examples

1.
```
SNANCP> SET LINE SYN-0 STATE OFF
SNANCP> SET LINE SYN-0 STATE ON CLOCK EXTERNAL DUPLEX HALF -
_SNANCP> SIGNAL NORMAL LOGGING WARNING NOTE "LINE L022B"
```

The first command sets the line SYN-0 to the OFF state. The second command modifies the values of STATE, CLOCK, DUPLEX, SIGNAL, LOGGING, and NOTE.

2.
```
SNANCP> SET LINE CQ-0 STATE OFF
SNANCP> SET LINE CQ-0 STATE ON PROTOCOL -
_SNANCP> CHANNEL BYTE FORCED BURST ENABLED
```

The first command sets the line CQ-0 to the OFF state. The second command modifies the values of PROTOCOL and FORCED BURST, and sets the line back ON.

SET LOGGING

The SET LOGGING command defines or changes which DECnet nodes events are logged to, and whether the messages are recorded in a file or at a logging monitor.

This command controls how and where logging events are reported. For information on which events are logged for a certain component, see the LOGGING parameter description under the SET command for LINE, CIRCUIT, PU, LU, and SERVER.

Format

SET logging-component parameter [...]

Logging Components

KNOWN LOGGING
Defines or modifies the specified parameters for all known logging.

LOGGING FILE
Defines or modifies the specified parameters for logging binary files.

LOGGING CONSOLE
Defines or modifies the specified parameters for dedicated logging consoles or text files.

LOGGING MONITOR
Defines or modifies the specified parameters for logging monitors. The logging monitor is OPCOM (OPerator's COMmunications facility).

Command Parameters

The following table lists the default values for logging parameters:

Parameter	Value	Default
NAME	*file-spec*	None
SINK NODE	*node-id*	Load host
KNOWN SINKS	None	Load Host

Parameter	Value	Default
STATE	ON or OFF	ON

NAME *file-spec*
Indicates a file or program to which the messages are logged. There is no default name for the LOGGING FILE component.

Events are always logged to OPCOM. If you use the SET LOGGING MONITOR NAME command, events are logged to the named monitor process in addition to OPCOM.

KNOWN SINKS
Specifies that the event logging parameters are to be changed for all sink nodes.

SINK NODE *node-id*
Specifies the DECnet host node that logs the event messages. By default, the sink node is the load host.

STATE *keyword*
Specifies whether event logging is ON or OFF for the specified sink.

By default, the Gateway logs events to the monitor process (OPCOM) on the load host. The default state of this sink is ON. For all other sinks (that is, for all sinks you explicitly defined) the default state is OFF.

Examples

1.
```
SNANCP> SET KNOWN LOGGING STATE OFF KNOWN SINKS
```
   This sets all logging on all sink nodes to OFF.

2.
```
SNANCP> SET LOGGING MONITOR SINK NODE NODEB STATE ON
```
   This command causes event messages to be logged by OPCOM on NODEB.

3.
```
SNANCP> SET LOGGING CONSOLE SINK NODE BAHMAT -
_SNANCP> NAME SYS$MANAGER:GATEWAY.LOG STATE ON
```
   This command causes event messages to be sent to the named "console" text file.

4.
```
SNANCP> SET LOGGING FILE SINK NODE BAHMAT -
_SNANCP> NAME SYS$MANAGER:GATEWAY.DAT STATE ON
```

This command causes event messages to be written to the named file in binary format.

SET LU

The SET LU command defines or changes parameters for an LU. The LU must have already been defined with a SET PU command.

If you create authorization entries or change authorization information for an LU that is currently in use, the authorization information does not affect the current session. The Gateway checks the new information the next time it attempts to allocate the LU.

The AUTHORIZATION component qualifier cannot be entered on the same command line as the LOGGING and NOTE parameters. Use a separate SNANCP command line to create authorization entries.

Format

SET lu-component [component-qualifier] parameter [...]

LU Components

KNOWN LUS
Defines or changes the specified parameter(s) for all known LUs.

LU *lu-id*
Defines or changes parameters for a specific LU. Enter the *lu-id* in the form *pu-id.nnn*; for example, SNA-0.1.

Component Qualifiers

AUTHORIZATION *auth-id*
Defines or changes a specific authorization entry for the specified LU(s).

KNOWN AUTHORIZATIONS
Defines or changes all known authorization entries for the specified LU(s).

Authorization entries allow you to control access to Gateway functions on an LU basis. The *auth-id* is a name you use to identify an authorization entry. The *auth-id* can be used with other SNANCP commands such as CLEAR and SHOW to manipulate the authorization entry. Enter the *auth-id* as a string of up to 16 alphanumeric characters; for example, NEW_AUTH.

All access control parameters except PASSWORD support the asterisk (*) wild-card character. A wild-card (*) in the pattern matches zero or more characters in the string against which it is compared. The comparison is not sensitive to the case of the characters.

You can define any or all of the following access control parameters for an authorization entry:

NODE *node-name*	Indicates the name of the DECnet node from which the user is trying to access the LU. The node name is a string of up to 6 characters. If this parameter is null, it matches any accessing node. (See Chapter 7 for information about cluster node names.)
USER *username*	Indicates the name of the remote user trying to access the LU. The username is a string of up to 32 characters. If this parameter is null, it matches any accessing username.
TERMINAL *term-name*	Indicates the name of the terminal from which the user is trying to access the LU. The terminal name is a string of up to 32 characters. If this parameter is null, it matches any accessing terminal name. (See Chapter 7 for information about naming conventions.)
PASSWORD *password*	Indicates the password assigned to the authorized user for this LU. The password is a string of up to 32 characters. If this field is null, the user does not have to supply a password. You cannot use the asterisk wild-card character (*) in the password field.

There is no authorization information by default.

Command Parameters

The following table lists the default values for LU parameters:

Parameter	Value	Default
LOGGING	FATAL,ERROR WARNING, or INFORMATIONAL	FATAL
NOTE	0-30 chars	No note

LOGGING [ENABLED] *keyword*

Specifies which levels of messages are recorded. The keyword is one of the following:

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

The level you specify sets the minimum level to be logged; all levels of greater severity will also be logged. For example, if you set logging to INFORMATIONAL, messages at all other levels will also be logged. Fatal event messages are always logged. For more information on event messages, severity levels, and logging commands, see Chapter 10.

NOTE *string*
This parameter lets you specify additional information (no more than 30 characters) that will be part of the SHOW LU SUMMARY display. Specify the text of your note within quotation marks.

Examples

1.
```
SNANCP> SET LU SNA-0.1 NOTE "IBM LU L022B01"
```

   The command adds a note about the IBM LU name to the SHOW SUMMARY display for LU SNA-0.1.

2.
```
SNANCP> SET LU SNA-0.1 AUTHORIZATION TEST_AE NODE BRETT -
_SNANCP> USER SYSTEM PASSWORD YOURPSWD
SNANCP> SET LU SNA-0.1 LOGGING ERROR
```

   The first command defines the authorization entry for access to the LU SNA-0.1. String values are provided for NODE, USER, and PASSWORD. Because no value is provided for TERMINAL, any value matches that field. The second command specifies parameters other than authorization parameters. This parameter (LOGGING) cannot be entered on the same command line with authorization information.

3.
```
SNANCP> SET PU SNA-0 LU LIST 1-128
SNANCP> SET KNOWN LUS AUTH NORTH NODE YUKON USER MUSKOX
SNANCP> SET PU LU LIST 1-255
```

   This example shows an easy way to add the same authorization entry to many LUs. The PU SNA-0 is defined to have 128 LUs. An authorization entry is created for all of the LUs. The LU list is then expanded to 255 LUs; however, only the first 128 have the authorization information.

SET PU

The SET PU command defines or changes PUs and their associated parameters. The SET PU command also defines LUs for a PU. Make sure that the PU is inactive before using SET PU to define any parameters (except COUNTER TIMER, NOTE, and LOGGING). To make a PU inactive, set the corresponding circuit to the OFF state.

Format

SET pu-component parameter [...]

PU Components

KNOWN PUS
Defines or changes the specified parameter(s) for all known PUs.

PU *pu-id*
Defines or changes the specified parameters for a specific PU. Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0.

Command Parameters

The following table lists the default values for PU parameters:

Parameter	Value	Default
CIRCUIT	SDLC-*n*	None
	CHAN-*n*	None
COUNTER TIMER	0-65535	0
LOGGING	FATAL,ERROR WARNING, or INFORMATIONAL	FATAL
LU [LIST]	List, each 1-255	None
NOTE	0-30 chars	No note
QUERY THRESHOLD	0-65535	0
SEGMENT SIZE	265-1482 bytes	265
SSCP ID	list, each 4 hex digits	None

CIRCUIT *circuit-id*

Allocates a circuit to the PU. The circuit you specify must already have been defined with the SET CIRCUIT command.

- When using SDLC circuits, enter the *circuit-id* in the form SDLC-*n*; for example, SDLC-0.
- When using channel-attached circuits, enter the *circuit-id* in the form CHAN-*n*; for example, CHAN-2.

This parameter lets you switch the same PU between different circuits. You must specify this parameter and provide a *circuit-id* before you can use a PU.

COUNTER TIMER *seconds*

Specifies how long (0–65535 seconds) the counter timer should run. Each time the counter timer expires, an event message recording all the counter values is logged to the DECnet host.

The default is 0, which means the timer is not running.

LOGGING [ENABLED] *keyword*

Specifies which levels of messages are recorded. The keyword is one of the following:

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

The level you specify sets the minimum level to be logged; all levels of greater severity will also be logged. For example, if you set logging to INFORMATIONAL, messages at all other levels will also be logged. Fatal event messages are always logged. For more information on event messages, severity levels, and logging commands, see Chapter 10.

LU [LIST] *lu-list*

Defines a range of LU numbers (1-255) to be associated with the PU. The LU numbers must correspond with the LU addresses defined by the setting of the LOCADDR parameter for the IBM ACF/NCP LU macro or the VTAM LU macro.

Separate each number or range of numbers by a comma. The list you specify replaces any LU list already defined for that PU. Note that the Gateway rejects any attempt to activate an LU not specified in the list. The following is a sample LU list:

```
1-6,8,10,12-15
```

NOTE *string*
Lets you specify additional information (no more than 30 characters) that will be part of the SHOW SUMMARY display. Specify the text of your note within quotation marks.

[QUERY] THRESHOLD *seconds*
Specifies the number of seconds (0–65535) that must elapse between REQMS (Request Maintenance Statistics) messages.

The IBM Network Problem Determination Application (NPDA) can send REQMS messages, asking for Gateway counters, at regular intervals. This parameter defines what the interval must be. For messages sent at the proper intervals, the Gateway sends its counters back to NPDA in a RECFMS (Record Formatted Maintenance Statistics) message. The Gateway returns a negative response to any messages not sent at the proper intervals. If the value of this parameter is 0, the Gateway processes all REQMS messages it receives. See Appendix B for information on REQMS messages.

The default is 0.

SEGMENT [SIZE] *bytes*
When using this parameter with SDLC circuits, it specifies the size of the frames (265-1482) to transmit to the IBM host. The SEGMENT SIZE value should match the value of the MAXDATA parameter for the IBM ACF/NCP PU macro. The LINE BUFFER SIZE parameter is related to SEGMENT SIZE.

The default is 265.

When using this parameter with channel-attached circuits, any value specified for this parameter is ignored. Instead, the Gateway-CT computes a suitable value from the VTAM connection parameters at the time the link is established.

SSCP [ID] *sscp-list*
Identifies the SSCPs that could activate this PU. You should specify no more than 8 SSCP IDs. Each *sscp-id* must be 4 hexadecimal digits long. Each list you specify becomes the new list for that PU.

An SSCP identifier must correspond with the IBM VTAM SSCPID start option specified for the VTAM hosts to which the Gateway can be connected.

The Gateway compares the SSCP ID in an SNA activation message with the SSCP IDs you have specified. If the SSCP identifier in the activation message matches any of the identifiers defined, the Gateway accepts the activation; otherwise the Gateway rejects it. If you do not define any SSCP identifiers, the Gateway accepts an activation message with any SSCP identifier.

Example

```
SNANCP> SET CIRCUIT SDLC-0 STATE OFF
SNANCP> SET PU SNA-0 CIRCUIT SDLC-0 SSCP ID 0040 -
_SNANCP> LU LIST 1-19 SEGMENT SIZE 8316 NOTE "IBM PU P022B"
```

The first command sets the circuit SDLC-0 to the OFF state. The associated PU, SNA-0, becomes inactive. The second command defines new values for CIRCUIT, SSCP ID, LU LIST, SEGMENT SIZE, and NOTE.

SET SERVER

The SET SERVER command defines or changes servers and their parameters.

Before defining a server or setting server parameters, make sure the server is in the OFF state.

Format

SET server-component [component-qualifier] parameter [...]

Server Components

KNOWN SERVERS
Defines or changes parameters for all known servers.

SERVER *server-id*
The server-id is one of the following:

- **DHCF**—Lets IBM 3270 type terminals using the IBM HCF program access DHCF on an OpenVMS host. For more information, see *DECnet SNA VMS Distributed Host Command Facility Management*.
- **RJE**—Lets DECnet SNA Remote Job Entry (RJE) for OpenVMS perform SNA RJE emulation. For more information, see the *DECnet SNA Remote Job Entry for OpenVMS User's and Operator's Guide*.
- **SNA-ACCESS**—Lets the remaining DECnet SNA OpenVMS access routines, such as the DECnet SNA Printer Emulator (PrE) for OpenVMS, the DECnet SNA Data Transfer Facility (DTF), DECnet SNA VMS DISOSS Document Exchange Facility, and DECnet SNA 3270 Terminal Emulator for OpenVMS access an SNA network.

For DHCF, set the server state to OFF before you set the maximum links and PU LU list parameters.

Component Qualifiers

The following component qualifiers apply only to the SNA-ACCESS server.

KNOWN LUs
Allocates all known LUs to the SNA-ACCESS server for outbound connection initiation.

LU *lu-id*

Allocates the specified LU(s) to the SNA-ACCESS server for outbound connection initiation.

You can define one or more of the following outbound connection parameters for the specified LU(s):

NODE *node-name*	Indicates the name of the DECnet node that will receive the BIND request. In a VAXcluster, you can use the actual node name or the alias node identifier for the entire cluster. The name that you use depends on the specific object to which you are connecting and your cluster environment. Refer to the documentation for the individual DECnet SNA OpenVMS access routines for information on how each layered product specifies a node name. Refer to the *OpenVMS Networking Manual* if you need more information on alias node identifiers.
OBJECT NAME *string*	A string of 1 to 12 characters that identifies the DECnet object that is to receive the BIND request.
OBJECT NUMBER *number*	A number from 0 to 255 that identifies the number of the DECnet object that is to receive the BIND request.

The following parameters define access control information that is used to establish a logical link to the DECnet node:

USER *username*	Indicates the user identification to establish the logical link. The username is a string of 1 to 39 characters.
PASSWORD *password*	Indicates the password to establish the logical link. The password is a string of 1 to 39 characters.
ACCOUNT *account*	Indicates the user account to establish the logical link. The account is a string of 1 to 39 characters.

The following table lists the default values for server parameters:

Parameter	Value	Default
LOGGING	FATAL,ERROR WARNING, or INFORMATIONAL	FATAL
MAXIMUM LINKS	1-1020	0
NOTE	0-30 chars	No note
STATE	ON or OFF	OFF
DHCF only:		
PU pu-id LU list	A list of LU numbers between 1 and 255.	None
TEXT FILE	*file-spec*	None

Command Parameters

LOGGING [ENABLED] *keyword*

Specifies which levels of messages are recorded. The keyword is one of the following:

- INFORMATIONAL
- WARNING
- ERROR
- FATAL

The level you specify sets the minimum level to be logged; all levels of greater severity will also be logged. For example, if you set logging to INFORMATIONAL, messages at all other levels will also be logged. Fatal event messages are always logged. For more information on event messages, severity levels, and logging commands, see Chapter 10.

MAXIMUM LINKS *number*

Specifies the maximum number (1-1020) of DECnet logical links that can be active for the server.

For SNA-ACCESS, this parameter limits the number of people who can use the access routines that need the SNA-ACCESS server. For the 3270 TE, for example, there must be one logical link for each 3270 TE user. For RJE, this parameter limits the number of concurrent workstations and streams. Each workstation needs one link, and each stream needs one link. For DHCF, this parameter limits the number of concurrent DHCF sessions.

If this parameter is not defined, a server-dependent limit is applied. Moreover, the maximum supported value depends on the Gateway system itself.

NOTE *string*
Lets you specify additional information (no more than 30 characters) that will be part of the SHOW SUMMARY display. Specify the text of your note within quotation marks.

STATE *keyword*
Specifies the operational state of the server as ON or OFF. The server state must be ON for you to use the Gateway functions.

The following parameters apply only to the DHCF server:

PU *pu-id* LU [LIST] *lu-list*
Allocates and deallocates LUs to DHCF. This parameter indicates which LUs are allocated to the DHCF server and with which PU the LUs are associated.

Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0. To specify an *lu-list*, the LU numbers must be between 1 and 255. Separate each number or range of numbers by a comma. The following is an example of an LU list:

```
1-6,8,10,12-15
```

For any given *pu-id*, SET SERVER PU *pu-id* LU [LIST] *lu-list* command replaces the existing LU list with the new list. To allows DHCF to use more than one PU, issue separate SET SERVER commands, with different PU *pu-id* LU [LIST] *lu-list* parameters.

The number of LUs allocated to DHCF cannot exceed the maximum number of logical links defined with the SET SERVER DHCF MAXIMUM LINKS command.

TEXT FILE *file-spec*
Specifies a file you want to display on the screen when a user connects to the Gateway node using HCF. You can edit the file to contain whatever information you wish. The contents of the file are translated to EBCDIC and displayed on lines 12–22 of the logon screen. The format for the file specification is as follows:

node"*access info*"*::device:[directory]filename.type*

The default node is the Gateway load host.

The same text file is used for the logon screen for all the LUs allocated to be used by DHCF users on the Gateway.

DHCF assumes that the 3270 screen has 80 columns. If a terminal is set to use more or less than 80 columns, the logon screen might not be displayed correctly. For information on setting the screen size, refer to the *DECnet SNA Gateway-ST Guide to IBM Parameters* or the *DECnet Gateway-CT Guide to IBM Parameters*.

Use the CLEAR SERVER TEXT FILE command to remove a current text file before you define a new one.

When you use the SET SERVER TEXT FILE command, DHCF reads the text file even if the server state is OFF. By looking at the SHOW SERVER DHCF CHARACTERISTICS display, you can ensure the logon screen has been included before you set the server state to ON.

Examples

1.
```
SNANCP> SET SERVER SNA-ACCESS STATE ON LOGGING ERROR
```

   This command defines the state of the SNA-ACCESS server as ON and sets the logging level to ERROR.

2.
```
SNANCP> SET SERVER DHCF STATE ON MAX LINKS 3 PU SNA-0 LU LIST 15-17 -
_SNANCP> TEXT FILE NODEA::DISK$DHCF:[DHCF]LOGON.TXT NOTE "LUS 15-17" -
_SNANCP> LOGGING WARNING LOGGING ERROR
```

   This command defines the STATE, MAXIMUM LINK, PU LU LIST, TEXT FILE, NOTE, and LOGGING parameters for the DHCF SERVER.

SHOW ACCESS NAME

The SHOW ACCESS NAME command displays information about the specified access name(s).

Format

SHOW access-name-component parameter [qualifier]

Access Name Components

ACCESS NAME *access-name-id*
Displays information for a particular access name. Enter the *access-name-id* as a string of 1 to 16 alphanumeric characters.

KNOWN ACCESS NAMES
Displays information for all known access names.

Command Parameters

CHARACTERISTICS
Displays information about access name definitions. The DATA value is shown only if you have entered the privileged username and password in the USE SYSTEM command. If you have entered the nonprivileged username and password, the DATA value is not displayed. Instead, you see the character string "*SET*" to indicate that a value has been given for this parameter.

SUMMARY
Displays existing access names and notes. This is the default for SHOW ACCESS NAME.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Example

```
SNANCP> SHOW KNOWN ACCESS NAMES
```

This command displays a list of all access names and any notes specified for the names.

SHOW CIRCUIT

The SHOW CIRCUIT command displays information about the specified circuit(s).

Format

SHOW circuit-component parameter [qualifier]

Circuit Components

ACTIVE CIRCUITS
Displays information for all active circuits.

KNOWN CIRCUITS
Displays information for all known circuits.

CIRCUIT *circuit-id*
Displays information for a particular circuit.

- When using SDLC circuits, enter *circuit-id* in the form SDLC-*n*; for example, SDLC-0.
- When using channel-attached circuits, enter *circuit-id* in the form CHAN-*n*; for example, CHAN-2.

Command Parameters

CHARACTERISTICS
Displays information about the parameters that have been set for the specified circuit(s).

COUNTERS
Displays information about circuit counters. See Appendix B for information about counters.

STATUS
Displays status information about the circuit(s). The display could contain only the circuit-id and state, or it could include additional information.

SUMMARY
Displays the circuit-id, state, and note. This is the default for SHOW CIRCUIT.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Examples

1. `SNANCP> SHOW CIRCUIT SDLC-0`

   This command produces a display similar to the following:

```
Circuit Summary as of 23-AUG-1988 10:45:18 from STAR

     Circuit        State          Note

     SDLC-0         on             IBM Line L022B
```

2. `SNANCP> SHOW CIRCUIT CHAN-1`

   This command produces a display similar to the following:

```
Circuit Summary as of 23-AUG-1988 10:45:18 from STAR

     Circuit        State          Note

     CHAN-1         on             Byte Multiplexer Channel
```

3. `SNANCP> SHOW CIRCUIT SDLC-0 COUNTERS`

   This command produces a display similar to the following:

```
Circuit Counters as of 21-AUG-1988 08:28:45 from STAR

Circuit = SDLC-0

        14729  Seconds since last zeroed
        14361  Bytes received
         1530  Bytes sent
          130  Data blocks received
          117  Data blocks sent
            0  Data errors inbound
            0  Data errors outbound
            0  Remote buffer errors
            0  Local buffer errors
            0  Remote station errors
            0  Local station errors
            0  Test frames received
            0  Test frames received, bad CRC
       931628  Polls received
            0  Frame retransmits
```

4. SNANCP> SHOW CIRCUIT CHAN-1 COUNTERS

   This command produces a display similar to the following:

```
Known Circuit Counters as of 19-AUG-1988 08:28:45 from STAR

Circuit = CHAN-1

        2236  Seconds since last zeroed
      328830  Bytes received
       12925  Bytes sent
         525  Data blocks received
         526  Data blocks sent
           4  Channel process errors, including:
                Data length error
                Connect rejected, invalid parameters
         231  Attentions sent
         312  Attentions sent on read
         307  Write channel programs
         531  Read channel programs
```

SHOW LINE

The SHOW LINE command displays information about the specified line(s).

Format

SHOW line-component parameter [qualifier]

Line Components

ACTIVE LINES
Displays information for all active lines.

KNOWN LINES
Displays information for all known lines.

LINE *line-id*
Displays information for a particular line.

- When using SDLC lines, enter the *line-id* in the form SYN-*n*; for example, SYN-0.
- When using channel-attached lines, enter the *line-id* in the form CQ-*n*; for example, CQ-0.

Command Parameters

CHARACTERISTICS
Displays information about the parameters that have been set for the specified line(s).

COUNTERS
Displays information about line counters. See Appendix B for information about counters.

STATUS
Displays status information for the specified line(s). The display could contain only the circuit-id and state, or it could include additional information.

SUMMARY
Displays the line identification, state, and note string for the specified line(s). This is the default for SHOW LINE.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Examples

1. `SNANCP> SHOW LINE CQ-0`

   This command produces a display similar to the following:

```
Line Summary as of 19-AUG-1988 14:20:49 from STAR

  Line            State                   Note

  CQ-0            on
```

2. `SNANCP> SHOW LINE SYN-0`

   This command produces a display similar to the following:

```
Line Summary as of 21-JUN-1988 14:11:51 from STAR

   Line             State                   Note

  SYN-0             on                      L1228-(V.35, 56kbps, SDLC)
```

3. `SNANCP> SHOW LINE SYN-0 COUNTERS`

   This command produces a display similar to the following:

```
Line Counters as of 21-AUG-1988 14:30:12 from STAR

Line = SYN-0

       2020 Seconds since last zeroed
     269779 Bytes received
     256689 Bytes sent
     127838 Blocks received
     127821 Blocks sent
          0 Carrier lost
          0 Process errors
          0 Receive failures
          0 Transmit failures
```

4. SNANCP> SHOW LINE CQ-0 COUNTERS

This command produces a display similar to the following:

```
Line Counters as of  5-FEB-1988 14:13:53 from STAR

Line = CQ-0

        6398  Seconds since last zeroed
        3569  Commands received
        3645  Channel operations executed
      675860  Bytes received
       30554  Bytes sent
        1362  Blocks received
        1357  Blocks sent
           4  Channel events, including:
                System reset
                Selective reset
           0  Process errors
           0  Channel errors
         389  Local buffer errors, including:
                Receive buffer ring empty
          64  Buffers in use at peak
           5  Attention buffers at peak
```

SHOW LOGGING

The SHOW LOGGING command displays information about the current logging parameters.

Format

SHOW logging-component parameter [qualifier]

Logging Components

KNOWN LOGGING
Displays information for all known logging.

LOGGING CONSOLE
Displays information for the logging console.

LOGGING FILE
Displays information for the logging file.

LOGGING MONITOR
Displays information for the logging monitor.

Command Parameters

CHARACTERISTICS
Displays information about logging parameters.

SINK NODE *node-id*
Specifies the sink node for which you want information.

KNOWN SINKS
Specifies that information is to be displayed for all known sink nodes. If a specific sink node is not specified, this parameter is the default.

STATUS
Displays status information about logging. This parameter could include information such as whether there is a LOGGING FILE or LOGGING MONITOR component.

SUMMARY
Displays summary information such as the type and the state of logging. This is the default for SHOW LOGGING.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Examples

1. `SNANCP> SHOW KNOWN LOGGING`

   This command displays information on the current logging. The SUMMARY display is the default.

2. `SNANCP> SHOW KNOWN LOGGING TO NODEB_LOGGING.LIS`

   This command puts the logging information into the file NODEB_LOGGING.LIS in the default directory.

3. `SNANCP> SHOW LOGGING MONITOR SINK NODE CANT`

   This command displays information about the sink node CANT and the state of logging.

SHOW LU

The SHOW LU command displays information about the specified LU(s).

Format

```
SHOW   lu-component [component-qualifier] parameter [qualifier]
```

LU Components

ACTIVE LUS
Displays information for all active LUs.

KNOWN LUS
Displays information for all known LUs.

LU *lu-id*
Displays information for the specified LU. Enter the *lu-id* in the form *pu-id.nnn*; for example, SNA-0.1.

Component Qualifiers

AUTHORIZATION *auth-id*
Displays information about the authorization entry you specify. Enter the *auth-id* as a string of up to 16 alphanumeric characters; for example, NEW_AUTH.

KNOWN AUTHORIZATIONS
Displays information about all known authorization entries.

If you use a component qualifier, you might use only the SUMMARY or CHARACTERISTICS parameter. For example, the following command displays summary information:

```
SNANCP> SHOW LU SNA-0.1 KNOWN AUTHORIZATION
```

The following command displays output on authorization characteristics (you must be a privileged user to use this command):

```
SNANCP> SHOW LU SNA-0.1 KNOWN AUTHORIZATION CHARACTERISTICS
```

Command Parameters

CHARACTERISTICS

Displays information about the levels of event logging or about authorization entries for the specified LU(s).

COUNTERS

Displays information about counters for the latest LU–LU session.The latest session can be either the current session or the last session. Counters from the latest session are automatically set to zero at the start of each new session. See Appendix B for information on counters.

STATUS

Displays the LU identification, LU status, and BIND parameters, if the specified LU is in any of the active states. The BIND parameters for an active LU can include any of the following:

Parameter	Format
LU type	*n*
Primary LU	text
RU size (in)	*nnnn*
RU size (out)	*nnnn*
TS profile	*nn*
FM profile	*nn*
Last receive check	*xxxx*
Last transmit check	*xxxx*
Secondary send pacing	*nn*
Secondary receive pacing	*nn*
Logon mode	text
User data	text

Note

The user data BIND parameter is displayed only if you use privileged access control.

The Last receive check shows the IBM status code for the last error that occurred while the Gateway was receiving information. The Last transmit check shows the IBM status code for the last error that occurred while the Gateway was transmitting information. Look up the sense code in the IBM *Systems Network Architecture Formats* manual. The Last receive check and Last transmit check parameters are suppressed when 0. Other BIND parameters can be suppressed when blank.

SUMMARY
Displays the following information for the specified LU:

- LU identification
- State
- Note (if one exists)
- Which server is connected to the LU (if applicable)

This SUMMARY display is the default for SHOW LU.

When the AUTHORIZATION or KNOWN AUTHORIZATION component qualifiers are used with the SUMMARY parameter, the summary display includes information about the authorization entries.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Example

```
SNANCP> SHOW LU SNA-0.1
```

This command displays the SUMMARY information.

SHOW PU

The SHOW PU command displays information about the specified PU(s).

Format

SHOW pu-component parameter [qualifier]

PU Components

ACTIVE PUS

Displays information for all active PUs.

KNOWN PUS

Displays information for all known PUs.

PU *pu-id*

Displays information for the specified PU. Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0.

Command Parameters

CHARACTERISTICS

Displays information about the parameters that have been set for the specified PU(s).

COUNTERS

Displays information about PU counters. The display includes information on the number of active LUs. The display contains information on SNA RECFMS messages. However, the counters do not contain the same information that is sent in RECFMS messages. For information on counters, see Appendix B.

STATUS

Displays status information for the specified PU(s). The display could contain only the PU identification and state, or it could include additional information.

SUMMARY

Displays the PU identification, state, and note string. This is the default for SHOW PU.

Qualifier

TO *file-spec*
Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Example

```
SNANCP> SHOW PU SNA-0
```

This command produces a display similar to the following:

```
PU Summary as of 21-AUG-1988 11:30:20 from STAR

      PU           State              Note

     SNA-0         inactive           IBM PU P022B LUs LO22B01
```

SHOW SERVER

The SHOW SERVER command displays information about the specified server(s).

Format

SHOW server-component [component-qualifier] parameter [qualifier]

Server Components

KNOWN SERVERS
Displays information for all servers.

SERVER *server-id*
Displays information for the specified server. The *server-id* is one of the following:

- DHCF
- RJE
- SNA-ACCESS

Component Qualifiers

The following component qualifiers apply only to the SNA-ACCESS server.

KNOWN LUS
Displays outbound connection initiation information for all LUs allocated to the SNA-ACCESS server.

LU *lu-id*
Displays outbound connection initiation information for the specified LU, which must be allocated to the SNA-ACCESS server.

If you use one of the component qualifiers with the SNA-ACCESS server, you might use only the SUMMARY or CHARACTERISTICS parameter. For example, the following command displays summary information:

```
SNANCP> SHOW SERVER SNA-ACCESS KNOWN LUS
```

The following command displays characteristics of LU SNA-0.1, which is allocated to the SNA-ACCESS server for outbound connections:

```
SNANCP> SHOW SERVER SNA-ACCESS LU SNA-0.1 CHARACTERISTICS
```

Command Parameters

CHARACTERISTICS

Displays information about the current characteristics of the specified server or about outbound connection initiation.

The SHOW SERVER DHCF CHARACTERISTICS display differs slightly from the display for the other servers.

COUNTERS

Displays information about the counters for the current session.

There are no SNANCP counters for the RJE server; counters for RJE are maintained by the host-based RJE multiplexer.

STATUS

Displays the server name and state. If applicable, this parameter also displays the corresponding DECnet logical link number and DECnet node name, as well as specific information about the connected LUs for each server. For SNA-ACCESS, the username and terminal name are also shown.

The SHOW SERVER STATUS display is different for each server.

SUMMARY

Displays the server name, state, and note text. This SUMMARY display is the default for SHOW SERVER. If you specify LU or KNOWN LUS, the summary display includes LU-specific information.

Qualifier

TO *file-spec*

Specifies an output file. If you do not specify a file, SYS$OUTPUT is the default.

Example

```
SNANCP> SHOW KNOWN SERVERS
```

This command produces a display similar to the following:

```
Known Server Summary as of 17-FEB-1988 10:01:17 from ALAS

  Server           State                  Note

 SNA-ACCESS        on                     Gateway Access Server
 DHCF              on                     Host Command Facility
 RJE               on                     Remote Job Entry
```

SPAWN

The SPAWN command suspends SNANCP and creates an OpenVMS subprocess. You can use this command to leave SNANCP temporarily to work at DCL level or to invoke OpenVMS utilities or layered products.

If you specify a DCL command as a parameter to SPAWN, the command is executed in the OpenVMS subprocess. When the command completes, the subprocess terminates, and control is returned to SNANCP.

If you do not enter a command string after SPAWN, press Return and you see the system prompt. You can then enter DCL commands, or you can invoke OpenVMS utilities or layered products. To return to SNANCP, log out of the subprocess.

Format

SPAWN [command-string]

Command Parameter

command-string
Specifies a DCL command to be executed in the context of the created subprocess. When the command completes, the subprocess terminates, and control is returned to the parent process.

Examples

1. `SNANCP> SPAWN MAIL`

   This command suspends your SNANCP process and creates a subprocess in which the MAIL utility is invoked. When you exit from MAIL, you are returned to your SNANCP process.

2. `SNANCP> SPAWN NCP TELL STAR SHOW KNOWN LINKS`

   This command suspends your SNANCP process and creates a subprocess in which NCP is invoked. After executing the TELL command, you are returned to your SNANCP process.

USE SYSTEM

The USE SYSTEM command specifies the Gateway node that is used for SNANCP commands. You specify this command immediately after invoking SNANCP or you cannot use any other SNANCP commands (except HELP, SPAWN, and EXIT). You do not need to specify this command if you have defined the logical name SNAGM$GATEWAY to provide a default Gateway node specification to SNANCP.

Format

USE SYSTEM node-name [access-information]

Command Parameter

node-name
This is the name of the Gateway node that is queried for information when you use subsequent SNANCP commands.

access-information
You can specify either privileged or nonprivileged access control information. A privileged username and password can be defined in the configuration command procedure during a Gateway installation, or the Gateway manager can define privileged access control information with the following command:

NCP>TELL *gateway-node* SET EXECUTOR PRIVILEGED USER *name* PASSWORD *password*

The Gateway manager can require nonprivileged access control information to use SNANCP by defining a nonprivileged username and password with the following command:

NCP>TELL *gateway-node* SET EXECUTOR NONPRIVILEGED USER *name* PASSWORD *password*

When you enter privileged or nonprivileged access information with the USE SYSTEM command, there are two possible forms:

USE SYSTEM *node-name* USER *username* PASSWORD *password*

USE SYSTEM *node-name*"*username password*"::

Examples

1. `SNANCP> USE SYSTEM STAR USER BRETT PASSWORD SECRET`

   This command specifies the Gateway node STAR as the executor node for SNANCP commands. The privileged username BRETT and privileged password SECRET allow you to use all SNANCP commands and to see all information in displays. If you do not enter the privileged access information, you can enter only the EXIT, HELP, SHOW, and SPAWN commands.

2. `SNANCP> USE SYSTEM STAR USER ALL PASSWORD NONPRIV`

   This command specifies the Gateway node STAR as the executor node for SNANCP commands. The nonprivileged username ALL and nonprivileged password NONPRIV allow you to use the EXIT, HELP, SHOW, and SPAWN commands. Some fields of the SHOW displays merely indicate SET for nonprivileged users.

   If a nonprivileged username and password are defined, you cannot use any of the SNANCP commands unless you specify the nonprivileged information.

ZERO CIRCUIT

The ZERO CIRCUIT command sets circuit counters to zero. If INFORMATIONAL logging is enabled, an event message recording the counters is logged to all the logging sinks (that is, logging monitors or files on all sink nodes).

The word COUNTERS is an optional part of the ZERO command.

Format

ZERO circuit-component *[COUNTERS]*

Circuit Components

CIRCUIT *circuit-id*
Sets the counters for a particular circuit to zero.

- When using SDLC circuits, enter the *circuit-id* in the form SDLC-*n*; for example, SDLC-0.
- When using channel-attached circuits, enter the *circuit-id* in the form CHAN-*n*; for example, CHAN-2.

KNOWN CIRCUITS
Sets the counters for all known circuits to zero.

Examples

1. `SNANCP> ZERO CIRCUIT SDLC-0 COUNTERS`

   This command resets all the counters for circuit SDLC-0.

2. `SNANCP> ZERO KNOWN CIRCUITS`

   This command resets all the counters for all known circuits.

ZERO LINE

The ZERO LINE command sets line counters to zero. If INFORMATIONAL logging is enabled, an event message recording the counters is logged to all the logging sinks (that is, logging monitors or files on all sink nodes).

The word COUNTERS is an optional part of the ZERO command.

Format

ZERO line-component *[COUNTERS]*

Line Components

KNOWN LINES
Sets the counters for all known lines to zero.

LINE *line-id*
Sets the counters for a particular line to zero.

- When using SDLC lines, enter *line-id* in the form SYN-*n*; for example, SYN-0.
- When using channel-attached lines, enter *line-id* in the form CQ-*n*; for example, CQ-0.

Examples

1. `SNANCP> ZERO LINE SYN-0 COUNTERS`

   This command resets all the counters for line SYN-0.

2. `SNANCP> ZERO KNOWN LINES`

   This command resets all the counters for all known lines.

ZERO LU

The ZERO LU command sets LU counters to zero. If INFORMATIONAL logging is enabled, an event message recording the counters is logged to all the logging sinks (that is, logging monitors or files on all sink nodes).

The word COUNTERS is an optional part of the ZERO command.

Format

ZERO lu-component *[COUNTERS]*

LU Components

KNOWN LUS
Sets the counters for all known LUs to zero.

LU *lu-id*
Sets the counters for a particular LU to zero. Enter the *lu-id* in the form *pu-id.nnn*; for example, SNA-0.1.

Example

```
SNANCP> ZERO LU SNA-0.8
```

This command resets all the counters for LU SNA-0.8.

ZERO PU

The ZERO PU command sets PU counters to zero. If INFORMATIONAL logging is enabled, an event message recording the counters is logged to all the logging sinks (that is, logging monitors or files on all sink nodes).

The word COUNTERS is an optional part of the ZERO command.

Format

ZERO pu-component *[COUNTERS]*

PU Components

KNOWN PUS
Sets the counters for all known PUs to zero.

PU *pu-id*
Sets the counters for a particular PU to zero. Enter the *pu-id* in the form SNA-*nnn*; for example, SNA-0.

Example

```
SNANCP> ZERO PU SNA-0
```

This command resets all the counters for PU SNA-0.

ZERO SERVER

The ZERO SERVER command sets server counters to zero. If INFORMATIONAL logging is enabled, an event message recording the counters is logged to all the logging sinks (that is, logging monitors or files on all sink nodes).

The word COUNTERS is an optional part of the ZERO command.

Format

ZERO   server-component *[COUNTERS]*

Server Components

KNOWN SERVERS
Sets the counters for all known servers to zero.

SERVER *server-id*
Sets the counters for a particular server to zero.

Example

```
SNANCP> ZERO SERVER SNA-ACCESS
```

This command resets all the counters for the SNA-ACCESS server.

14

NCP Commands

This chapter describes NCP commands used with the Gateway. The command descriptions give complete syntax and usage information for each command. Commands are in alphabetical order.

The NCP commands apply to both the DECnet SNA Gateway-ST and the DECnet SNA Gateway-CT. Any restrictions on use are noted in the Command Parameter section for a particular command.

CLEAR CIRCUIT

The CLEAR CIRCUIT command resets circuit parameters to their default values.

You use this command when the Gateway is the executor.

Format

CLEAR circuit-component parameter [...]

Circuit Components

ACTIVE CIRCUITS
Indicates that parameters for all active circuits are to be updated.

CIRCUIT *circuit-id*
Resets the parameters for the specified circuit to default values. Enter the *circuit-id* in the form ETHERNET.

KNOWN CIRCUITS
Resets the parameters for all known circuits to default values.

Command Parameters

COUNTER TIMER *seconds*
Cancels the logging timer. This cancellation prevents any further circuit counter logging for the specified circuit(s).

HELLO TIMER *seconds*
Resets the HELLO TIMER to the default value of 15 seconds.

Example

```
NCP>CLEAR CIRCUIT ETHERNET HELLO TIMER
```

This command resets the Ethernet circuit HELLO TIMER to the default value of 15 seconds.

CLEAR EXECUTOR

The CLEAR EXECUTOR command resets parameters to the default values (if any). CLEAR EXECUTOR ALL removes parameters from the volatile database on the Gateway node.

You use this command when the Gateway is the executor.

Format

CLEAR EXECUTOR parameter [...]

Command Parameters

ALL
Removes from the database all executor parameters.

COUNTER TIMER
Removes from the volatile database the logging timer to prevent any further circuit counter logging for the local node.

IDENTIFICATION
Removes from the volatile database the identification string for the local node.

NONPRIVILEGED *item*
Removes from the volatile database the specified nonprivileged access control information. Specify either or both of the following items for removal:

PASSWORD
USER

PRIVILEGED *item*
Removes from the volatile database the specified privileged access control information. Specify either or both of the following items for removal:

PASSWORD
USER

Example

```
NCP>SET EXECUTOR NODE THYME
NCP>CLEAR EXECUTOR COUNTER TIMER
```

The first command makes the Gateway node the executor node. The second command removes the counter timer value from the volatile database on the Gateway node.

CLEAR EXECUTOR NODE

The CLEAR EXECUTOR NODE command clears the default executor designation for all NCP commands. The executor of commands becomes the local node. The TELL prefix cannot be used with this command.

After you issue the CLEAR EXECUTOR NODE command, NML performs all NML operations. NML uses the privileges of your current process instead of the default nonprivileged or privileged accounts.

This command causes the executor to become the local node.

Format

```
CLEAR EXECUTOR NODE
```

Command Parameters

None

Example

```
NCP>SET EXECUTOR NODE THYME"SYSTEM SECRET"
    .
    .
    .
NCP>CLEAR EXECUTOR NODE
```

The first command sets the executor node to node THYME. The second command clears the default executor node designation specified previously by the SET EXECUTOR NODE command.

CLEAR LINE

The CLEAR LINE command resets parameters to the default values.

You use this command when the Gateway is the executor.

Format

CLEAR line-component parameter [...]

Line Components

KNOWN LINES
Resets the specified values for all known lines to the default values.

LINE *line-id*
Identifies the specific line whose parameters are to be either reset to their default values (if any) or removed from the volatile database.

Command Parameter

COUNTER TIMER *seconds*
Cancels the logging timer. This cancellation prevents any further line counter logging for the specified line(s).

Example

```
NCP>CLEAR LINE ETHERNET COUNTER TIMER
```

This command resets the COUNTER TIMER to the default. This cancels line counter logging.

CLEAR LOGGING EVENTS

The CLEAR LOGGING EVENTS command removes source-related logging parameters from the volatile database on the Gateway node.

You use this command when the Gateway is the executor.

Format

CLEAR logging-component parameter [...]

Logging Components

KNOWN LOGGING
Removes from the volatile database the parameters for all known logging.

LOGGING CONSOLE
Removes from the volatile database the parameters for logging to a console.

LOGGING FILE
Removes from the volatile database the parameters for logging to a file.

LOGGING MONITOR
Removes from the volatile database the parameters for logging to a monitor program.

Command Parameters

EVENT *event-list*
KNOWN EVENTS
Removes from the volatile database an event class and type or all event classes and types. You can associate the following source parameter with the EVENTS parameter.

SINK NODE *node-id*
Identifies the node at which events are being logged. The default node is the load host.

Example

```
NCP>CLEAR LOGGING CONSOLE SINK NODE NODEA KNOWN EVENTS
```

This command clears logging to the logging console on NODEA.

CLEAR NODE

The CLEAR NODE command removes node parameters from the volatile database on the Gateway node.

You use this command when the Gateway is the executor.

Format

CLEAR node-component parameter [...]

Node Component

NODE *node-id*
Identifies the node (local or remote) whose parameters are to be removed from the volatile database on the Gateway node.

Command Parameters

ALL
Removes all parameters for the specified node or all known nodes from the volatile database. The component is no longer recognized by the network.

COUNTER TIMER
Removes the COUNTER TIMER value from the volatile database.

NAME
Removes the node's name from the volatile database.

Example

```
NCP>CLEAR NODE TRNTO ALL
```

This command removes all parameter entries for node TRNTO in the volatile database. As a result, the node no longer exists for the executor node.

CONNECT NODE

The CONNECT NODE command sets up a logical connection between the host node and the console interface on a specified target node. Both the host node and the target node must be on the same Ethernet. This command is executed on the node running NCP, regardless of any prior SET EXECUTOR NODE command.

To define the default information for the target node in the issuing node's volatile database, use the NCP command SET NODE to specify the SERVICE CIRCUIT, and HARDWARE ADDRESS parameters for the target node. Once set, you can override these default parameters for the target node by specifying new parameters in the CONNECT command.

You use this command when the load host is the executor.

Format

CONNECT   node-component parameter [...]

Node Component

NODE *node-id*
Identifies the node name or address of the target node to be connected to the local node

Command Parameters

PHYSICAL ADDRESS *E-address*
Specifies the Ethernet physical address of the target node. The value is the Ethernet address that the target node has set for itself or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter associated with the target node in the executor node's volatile database.

SERVICE PASSWORD *hex-password*
Identifies the password required to create the logical link between the host node and the target node. The password is a hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF.

VIA *circuit-id*
Specifies the circuit to be used to create the logical link between the host node and the target node. The circuit must be an Ethernet circuit.

Example

```
NCP>CONNECT NODE RTRDEV SERVICE PASSWORD FFFFFFFFFFFFFFFF-
_VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-38-00
```

This command connects the host node to the console interface on the target node RTRDEV specifying the service password FFFFFFFFFFFFFFFF, the service circuit UNA-0, and the Ethernet physical address AA-00-04-00-38-00.

CONNECT VIA

The CONNECT VIA command sets up a logical connection between the host node and the console interface on a target node using the specified circuit. If the DECnet node address of the target node is not known, use the CONNECT VIA command instead of the CONNECT NODE command. Both the host node and the target node must be on the same Ethernet circuit. This command is executed on the node running NCP, regardless of any prior SET EXECUTOR NODE command.

This command is used when the load host is the executor.

Format

CONNECT VIA circuit-id parameter [...]

Command Parameters

PHYSICAL ADDRESS *E-address*
Specifies the Ethernet physical address of the target node. The value is the Ethernet address that the target node has set for itself. This parameter is required.

SERVICE PASSWORD *hex-password*
Identifies the password required to create the logical link between the host node and the target node. The password is a hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF.

Example

```
NCP>CONNECT VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-38-00
```

This command connects the host node to the console interface on the target node by specifying the circuit over which the connection is to be made and the Ethernet physical address of the target node.

HELP

Use the HELP command to obtain general information about NCP commands and parameters.

Format

HELP topic

Command parameter

topic
Gives information about a command you supply. The command words that can be used as topics are listed in the HELP display under Information Available.

Example

```
NCP>HELP LOOP NODE
```

This command provides information on the LOOP NODE command.

LOAD NODE

The LOAD NODE command down-line loads software to a target node, using either the identified circuit (if specified) or the circuit obtained from the volatile database. Any parameter left unspecified in the command defaults to whatever entry is specified in the volatile database on the executor node.

You use this command when the load host is the executor.

Note

This command is not supported when using the Gateway-CT; if you use the LOAD NODE command it will act as a TRIGGER NODE command.

Format

LOAD node-component parameter [...]

Node Components

NODE *node-id*
Identifies the node name or address of the target node to be loaded.

Command Parameters

ADDRESS *node-address*
Identifies the address that the target node uses.

FROM *file-spec*
Identifies the file specification of the load file containing the system software to be loaded.

HOST *node-id*
Identifies the default host that the target node uses.

NAME *node-name*
Identifies the name that the target node uses.

PHYSICAL ADDRESS *E-address*
Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node to be loaded currently uses to identify itself. The value is the Ethernet address that the target node has set for itself or, if the target

node has not set an Ethernet address, the HARDWARE ADDRESS parameter associated with the target node in the executor node's volatile database.

SERVICE PASSWORD *hex-password*
Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a hexadecimal number ranging from 0 to FFFFFFFFFFFFFFFF.

VIA *circuit-id*
Identifies the circuit over which the loading sequence is to take place.

Examples

1. `NCP>LOAD NODE NYC HOST BOSTON VIA UNA-0`

   This command initiates a load operation for node NYC over circuit UNA-0. When loaded, node NYC has node BOSTON as its default host specification.

2. `NCP>LOAD NODE BANGOR`

   This command initiates a load operation for node BANGOR. Any required default information is retrieved from the volatile database on the executor node.

3. `NCP>LOAD NODE SWIFT PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command initiates a load operation for node SWIFT on an Ethernet circuit. The executor node uses the Ethernet physical address specified in the command to address node SWIFT.

LOAD VIA

The LOAD VIA command loads software to a target node using the specified circuit. The target node identification is obtained from the volatile database on the executor node. If the target node is on an Ethernet circuit, you must specify the PHYSICAL ADDRESS parameter in this command.

This command is used when the load host is the executor.

Note

LOAD VIA is not supported when using the Gateway-CT; if you use the LOAD VIA command it will act as a TRIGGER VIA command.

Format

LOAD VIA circuit-id parameter [...]

Command Parameters

ADDRESS *node-address*
Identifies the address that the target node uses.

CPU *cpu-type*
Identifies the node's CPU type.

FROM *file-spec*
Identifies the file specification of the load file containing the system software to be loaded.

HOST *node-id*
Identifies the default host that the target node uses.

NAME *node-name*
Identifies the name that the target node uses.

PHYSICAL ADDRESS *E-address*
Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node to be loaded currently uses to identify itself. The value is the Ethernet address that the target node has set for itself or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter if associated with the target node in the executor node's volatile database.

Examples

1. `NCP>LOAD VIA BNT-0`

   This command initiates a load operation over circuit BNT-0. Any required default information is retrieved from the volatile database on the executor node. NCP scans the node database until it finds a node whose service circuit matches the load circuit.

2. `NCP>LOAD VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command initiates a load operation over Ethernet circuit UNA-0 to the target node whose Ethernet physical address is specified in the command.

LOOP CIRCUIT

The LOOP CIRCUIT command initiates a circuit level loopback test over the specified circuit. See *DECnet SNA Gateway-ST Problem Solving* or *DECnet SNA Gateway-CT Problem Solving* for additional information on loopback tests.

This command is used when the load host (or any other DECnet node) is the executor.

Format

LOOP   circuit-component parameter [...]

Circuit Component

CIRCUIT *circuit-id*
Identifies the circuit to be used for the loopback test.

Command Parameters

COUNT *number*
Specifies the number of blocks to be sent over the circuit during loopback testing. The count must be a decimal integer from 1 through 65535. If the parameter is omitted, only one block is looped.

LENGTH *number*
Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer ranging from 20 through 1470. If the parameter is omitted, a block length of 40 bytes is used.

WITH *data-type*
Specifies the type of binary information to be sent during testing. If the parameter is omitted, a combination of ones and zeros (the MIXED data type) is sent.

Ethernet Parameters

ASSISTANT NODE *node-id*
Identifies the name or address of the node that performs the role of loopback assistant for Ethernet third-party loop testing. This parameter can be used instead of the ASSISTANT PHYSICAL ADDRESS parameter.

ASSISTANT PHYSICAL ADDRESS *E-address*
Identifies the Ethernet physical address of the node that performs the role of loopback assistant for Ethernet third-party loop testing. ASSISTANT PHYSICAL ADDRESS must be specified if HELP is included in this command. The address cannot be a multicast address. This parameter can be used instead of the ASSISTANT NODE parameter.

HELP *help-type*
Indicates the type of assistance to be provided during Ethernet loopback testing by the assistant node, whose address is specified in the ASSISTANT PHYSICAL ADDRESS or whose node-id is specified in the ASSISTANT NODE parameter. There are three possible values for help-type:

TRANSMIT	The assistant node relays the request to the destination node, which replies directly to the executor node.
RECEIVE	The executor node sends the request directly to the destination node, which relays the reply to the assistant node for transmission to the executor node.
FULL	The assistant node relays the request and the reply between the executor node and the destination node.

If HELP is specified, ASSISTANT PHYSICAL ADDRESS or ASSISTANT NODE must also be specified.

NODE *node-id*
Applies only to Ethernet circuits. Identifies the destination node to be used for loopback testing of the specified Ethernet circuit. You can use this parameter instead of the PHYSICAL ADDRESS parameter.

PHYSICAL ADDRESS *E-address*
Applies only to Ethernet circuits. Identifies the Ethernet physical address of the destination node to be used for loopback testing of the specified Ethernet circuit.

Example

```
NCP>LOOP CIRCUIT ETHERNET NODE 224
```

This command initiates a circuit level loopback test with a node whose address is 224, over the Ethernet circuit.

LOOP EXECUTOR

The LOOP EXECUTOR command tests the executor node in the network by causing test data to be transmitted to the executor node. The parameters are optional and can be entered in any order. You can supply explicit access control information for the LOOP EXECUTOR command. See *DECnet SNA Gateway-ST Problem Solving* or *DECnet SNA Gateway-CT Problem Solving* for more information on loopback tests.

This command is used when the load host (or any other DECnet node) is the executor.

Format

LOOP EXECUTOR [parameter] [...]

Command Parameters

COUNT *number*
Specifies the number of blocks to be sent over the executor node during loopback testing. The count must be a decimal integer from 1 through 65535. If the parameter is omitted, only one block is looped.

LENGTH *number*
Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer ranging from 1 to 65535. If the parameter is omitted, a block length of 40 bytes is used.

WITH *data-type*
Specifies the type of binary information to be sent during testing. If this parameter is omitted, a combination of ones and zeros (the MIXED data type) is sent. The three types of data that can be sent are as follows:

MIXED
ONES
ZEROS

Example

```
NCP>LOOP EXECUTOR
```

This command initiates a loopback test on the executor node.

LOOP NODE

The LOOP NODE command initiates a node level loopback test.

To test the executor node, use the LOOP EXECUTOR command.

This command is used when the load host (or any other DECnet node) is the executor.

Format

LOOP node-component parameter [...]

Node Component

NODE *node-id*
Specifies the name of the remote node to be used in the loopback testing.

Command Parameters

COUNT *number*
Specifies the number of blocks to be sent over the node during loopback testing. The count must be a decimal integer from 1 through 65535. If the parameter is omitted, only one block is looped.

LENGTH *number*
Specifies the length (in bytes) of the blocks to be sent during loopback testing. The length must be a decimal integer ranging from 1 through 65535. If the parameter is omitted, a block length of 40 bytes is used.

WITH *data-type*
Specifies the type of binary information to be sent during testing. If this parameter is omitted, a combination of ones and zeros (the MIXED data type) is sent. The three types of data that can be sent are as follows:

MIXED
ONES
ZEROS

Example

```
NCP>SET NODE TESTER CIRCUIT ETHERNET
NCP>LOOP NODE TESTER
    .
    .
    .
NCP>CLEAR NODE TESTER CIRCUIT
```

The first command creates a loop node (TESTER) for the associated circuit, the second command initiates a node level loopback test with the loop node name, and the third command removes the loop node name from the volatile database when the loop test is completed.

SET CIRCUIT

The SET CIRCUIT command creates or modifies circuit parameters in the Gateway database.

You use this command when the Gateway is the executor.

Format

SET circuit-component parameter [...]

Circuit Components

CIRCUIT *circuit-id*
Defines or changes parameters for the circuit you specify. Enter the *circuit-id* in the form ETHERNET.

KNOWN CIRCUITS
Defines or changes the specified parameters for all known circuits.

Command Parameters

COUNTER TIMER *seconds*
Specifies the number of seconds that the circuit counter timer will run. When the counter timer expires, a circuit counter logging event occurs. Seconds must be a decimal integer in the range of 1 to 65535. If no value is set for COUNTER TIMER, the circuit counters are not logged automatically.

HELLO TIMER *seconds*
Specifies the frequency of routing hello messages sent to adjacent nodes on the circuit. Seconds must be a decimal integer in the range of 1 to 8191. The default value is 15.

Example

```
NCP>SET CIRCUIT ETHERNET HELLO TIMER 10
```

This command sets the Ethernet circuit hello timer to 10 seconds.

SET EXECUTOR

The SET EXECUTOR command defines or modifies parameters for the executor node.

You use this command when the Gateway is the executor.

Format

SET EXECUTOR   parameter [...]

Command Parameters

BUFFER SIZE *bytes*
Specifies the size of the line buffers and thereby controls the maximum segment size, including transport header, of all NSP messages received. The size must be in the range of 576 to 5000 bytes.

COUNTER TIMER *seconds*
Specifies a timer whose expiration causes a node counter logging event. The value must be a decimal integer in the range of 1 to 65535. If no value is set for COUNTER TIMER, the circuit counters are not logged automatically.

DELAY FACTOR *number*
Specifies the number by which to multiply one-sixteenth of the estimated round- trip delay to a node to set the retransmission timer to that node. Use a number in the range of 1 to 255. If you do not set this parameter, the default value is 80.

DELAY WEIGHT *number*
Specifies the weight to apply to a current round-trip delay data point when updating the estimated round trip-delay to a node. Use a number in the range of 1 to 255. If you do not set this parameter, the default value is 5.

IDENTIFICATION *id-string*
Is a text string that describes the executor node. The string can be a maximum of 32 characters. If it contains blanks or tabs, you must enclose the string in quotation marks.

INACTIVITY TIMER *seconds*
Specifies the maximum duration of inactivity (no data in either direction) on a logical link before the node checks to see if the logical link still works. The value range is 1 through 65535. If you do not set this parameter, the default value is 60.

MAXIMUM BUFFERS *number*
Specifies the maximum number of buffers in the transmit buffer pool. DECnet normally allocates only what it needs. At a minimum, use 15. Increase this value if you experience congestion loss. The default value is 500.

NONPRIVILEGED *item*
Specifies nonprivileged inbound access control information for the node. Associate either or both of the following items with the NONPRIVILEGED parameter:

PASSWORD *password*	Specifies the password for the nonprivileged account on the executor node.
USER *user-id*	Specifies the username for the nonprivileged account on the executor node.

The Gateway's handling of this parameter differs from that of an OpenVMS executor.

OUTGOING TIMER *seconds*
Specifies a time-out value for the elapsed time between the moment a connection is requested and the moment that connection is acknowledged by the destination node. It is recommended that you use a value from 30 to 60 seconds. The default value is 45.

PRIVILEGED *item*
Specifies privileged inbound access control information for the node. Associate either or both of the following items with the PRIVILEGED parameter:

PASSWORD *password*	Specifies the password for the privileged account on the executor node.
USER *user-id*	Specifies the username for the privileged account on the executor node.

The Gateway's handling of this parameter differs from that of an OpenVMS executor.

RETRANSMIT FACTOR *number*
Defines the maximum number of times any given message (except a connect initiate message) is retransmitted before the logical link is disconnected. If you do not set this parameter, the default is 10.

SEGMENT BUFFER SIZE *number*
Specifies in bytes the maximum size of the transmit buffers, thereby controlling the maximum size NSP message segment that can be transmitted. (This value is the maximum size message the End Communications layer can transmit; it

does not include Routing layer or Data Link layer overhead.) Use a value from 242 to 576. The default value is equal to the value of the BUFFER SIZE, if specified; otherwise, the default is 576.

The SEGMENT BUFFER SIZE is always less than or equal to the BUFFER SIZE. The two values are normally equal but could differ to permit the network manager to alter buffer sizes on all nodes without interruption of service.

Example

```
NCP>SET EXECUTOR OUTGOING TIMER 10
```

This command indicates that the executor node waits 10 seconds between the time a connection request is sent and the time it takes for an acknowledgment to be sent by the destination node.

SET EXECUTOR NODE

The SET EXECUTOR NODE command sets the default executor for all NCP commands. The executor is the node on which the Network Management Listener (NML) runs to perform these commands.

This command causes the Gateway to become the executor.

Format

SET EXECUTOR parameter [...]

Command Parameters

NODE *node-id*

Specifies a node name or address followed optionally by access control information as specified for OpenVMS.

The node identification could be a logical name. It is possible to override access control in a logical name with explicit access control information in the command.

When using the Gateway, this node-id identifies the Gateway node. The SET EXECUTOR NODE command therefore causes subsequent commands to be executed by the Gateway.

Associate either or both of the following optional access control parameters with the NODE parameter:

PASSWORD *password*	Specifies the user's password for access control verification at the designated executor node.
USER *user-id*	Specifies the user's identification for access control verification at the designated executor node.

Use either one of the following forms to enter access control information:

SET EXECUTOR NODE *node-id"user-id password"*

or

SET EXECUTOR NODE *node-id* USER *user-id* PASSWORD *password* ACCOUNT *account*

Examples

1. ```
   NCP>SET EXECUTOR NODE 5.14
       .
       .
       .
   NCP>CLEAR EXECUTOR NODE
   ```

   The first command sets the executor to node 5.14. The second resets the executor to the local node.

2. ```
   NCP>SET EXECUTOR NODE 14"GRAY MARY"
   ```

   This command uses access control information to set the executor node to node 1.14. This example assumes a default area number of 1 for the executor node.

SET LINE

The SET LINE command creates or modifies line parameters in the Gateway database.

You use this command when the Gateway is the executor.

Format

SET line-component parameter [...]

Line Components

KNOWN LINES
Resets parameters to the default values (if any) or removes the parameters for all known lines from the volatile database.

LINE *line-id*
Resets parameters to the default values (if any) or removes the parameters for the specified line from the volatile database.

Command Parameters

COUNTER TIMER *seconds*
Specifies the number of seconds that the line counter timer will run. When the counter timer expires, a line counter logging event occurs. Seconds must be a decimal integer in the range 1 to 65535. If no value is set for COUNTER TIMER, the line counters are not logged automatically.

RECEIVE BUFFERS *number*
Specifies the length of the line's receive queue. Use a value in the range of 2 to 128. The default is 64.

Example

```
NCP>SET LINE ETHERNET RECEIVE BUFFERS 8
```

This command sets the number of receive buffers to 8.

SET LOGGING EVENTS

The SET LOGGING EVENTS command creates or modifies source-related logging parameters in the volatile database on the local node. Source-related and sink-related parameters are mutually exclusive; you cannot use parameters from both categories in a single command.

You use this command when the Gateway is the executor.

Format

SET logging-component parameter [...]

Logging Components

KNOWN LOGGING
Creates or modifies the specified parameters for all known logging in the database.

LOGGING CONSOLE
Creates or modifies the specified parameters for the logging console in the database.

LOGGING FILE
Creates or modifies the specified parameters for the logging file in the database.

LOGGING MONITOR
Creates or modifies the specified parameters for the logging monitor in the database.

Command Parameters

EVENTS *event-list*
KNOWN EVENTS
Indicates a specific event or all known events to be logged. This parameter could be followed by the following parameter:

SINK NODE *node-id*
Identifies the node that is to receive events. The default node is the load host.

Refer to the *OpenVMS Networking Manual* for a table of the source-related parameters used with the SET LOGGING EVENTS command. For a summary of event class and types and information about specific events that OpenVMS logs, refer to the Supplemental NCP Information Section.

Examples

1. `NCP>SET LOGGING MONITOR KNOWN EVENTS`

   This command causes all events to be logged on the console.

2. `NCP>SET LOGGING CONSOLE KNOWN EVENTS SINK NODE TRNTO`

   This command causes all events generated by the Gateway to be logged to the logging console on remote node TRNTO.

SET NODE

The SET NODE command creates or modifies node parameters in the database on the executor.

You use this command when the Gateway is the executor.

Format

SET node-component parameter [...]

Node Components

KNOWN NODES
Creates or modifies the specified parameters in the database for all known nodes.

NODE *node-id*
Creates or modifies the specified parameters in the database for the local (or remote) node.

Command Parameters

COUNTER TIMER *seconds*
Specifies the number of seconds that the circuit counter timer will run. When the counter timer expires, a circuit counter logging event occurs. Seconds must be a decimal integer in the range 1 to 65535. If no value is set for COUNTER TIMER, the circuit counters are not logged automatically.

NAME *node-name*
Specifies the node name to be associated with the node identification. Only one name can be assigned to a node address.

Example

```
NCP>SET NODE 5.14 NAME TRNTO
```

This command assigns the node name TRNTO to node 5.14.

SHOW CIRCUIT

The SHOW CIRCUIT command displays the current circuit parameters.

You use this command when the Gateway is the executor.

Format

SHOW   circuit-component parameter [qualifier]

Circuit Components

ACTIVE CIRCUITS
Displays the parameters for all reachable (active) circuits connected to the local node.

CIRCUIT *circuit-id*
Displays the parameters for the specified circuit.

KNOWN CIRCUITS
Displays the parameters for all known circuits connected to the local node.

Command Parameters

CHARACTERISTICS
Displays parameters that are currently set for the circuit.

COUNTERS
Displays traffic information for the circuit. The COUNTERS parameter also tabulates recorded error conditions.

STATUS
Displays the availability of the circuit for network activity.

SUMMARY
Displays a summary of the parameters set for the circuit. SUMMARY is the default display type.

Qualifier

TO file-spec
Specifies the output file. If no output file is specified, SYS$OUTPUT is the default.

Example

```
NCP>SHOW CIRCUIT ETHERNET COUNTERS
```

This command displays the current counters for the Ethernet circuit.

SHOW EXECUTOR

The SHOW EXECUTOR command displays the current parameters for the executor node.

You use this command when the Gateway is the executor.

Format

SHOW EXECUTOR parameter [qualifier]

Command Parameters

CHARACTERISTICS
Displays parameter values that are currently set for the executor node.

COUNTERS
Displays traffic information for the executor node. The COUNTERS parameter also tabulates recorded error conditions.

STATUS
Displays the availability of the executor for network activity.

SUMMARY
Displays a summary of the parameters set for the executor node. SUMMARY is the default display type.

Qualifier

TO file-spec
Specifies the output file. If no output file is specified, SYS$OUTPUT is the default.

Examples

1. `NCP>SHOW EXECUTOR STATUS TO STAT.NOD`

   This command directs the status information for the executor node to an output file named STAT.NOD.

2. `NCP>SHOW EXECUTOR COUNTERS`

   This command displays the counters set for the executor node.

SHOW LINE

The SHOW LINE command displays the current line parameters.

You use this command when the Gateway is the executor.

Format

SHOW line-component parameter [qualifier]

Line Components

ACTIVE LINES
Displays information for all reachable (active) lines.

KNOWN LINES
Displays information for all known lines.

LINE *line-id*
Displays information for the specified line.

Command Parameters

CHARACTERISTICS
Displays parameters that are currently set for the line.

COUNTERS
Displays traffic information for the line. The COUNTERS parameter also tabulates recorded error conditions.

STATUS
Displays the availability of the line for network activity.

SUMMARY
Displays summary information about the specified line.

Qualifier

TO file-spec
Specifies the output file. If no output file is specified, SYS$OUTPUT is the default.

Examples

1. `NCP>SHOW ACTIVE LINES CHARACTERISTICS`

   This command displays line characteristics for all active lines, that is, lines set to ON. This display includes line parameters that you have set for individual lines.

2. `NCP>SHOW KNOWN LINES STATUS`

   This command displays status information for all known lines connected to the local node. This display includes the current state of the line.

SHOW LOGGING

The SHOW LOGGING command displays logging information for the executor node.

You use this command when the Gateway is the executor.

Format

SHOW logging-component parameter [qualifier]

Logging Components

KNOWN LOGGING
Displays information for all known logging.

LOGGING CONSOLE
Displays information about logging to a console.

LOGGING FILE
Displays information about logging to a file.

LOGGING MONITOR
Displays information about logging to a terminal.

Command Parameters

CHARACTERISTICS
Displays parameters currently set for logging.

EVENTS
Displays event logging information.

STATUS
Displays information about the state of logging.

SUMMARY
Displays summary information about logging.

Qualifier

TO file-spec
Specifies the output file. If no output file is specified, SYS$OUTPUT is the default.

Example

```
NCP>SHOW LOGGING CONSOLE CHARACTERISTICS SINK NODE TRNTO
```

This command displays characteristics for logging to a console on remote node TRNTO. This display includes both the sink node for which the events apply and those events that are set for the logging console component at the local node.

SHOW NODE

The SHOW NODE command displays remote node information on the executor node.

You use this command when the Gateway is the executor.

Format

SHOW node-component parameter [qualifier]

Node Components

ACTIVE NODES
Displays information for all reachable nodes.

KNOWN NODES
Displays information for all known nodes.

NODE *node-id*
Displays information for the specified node.

Command Parameters

CHARACTERISTICS
Displays information about node parameter values.

COUNTERS
Displays node error and performance statistics.

STATUS
Displays information about the current state of the node.

SUMMARY
Displays summary information about the node. This is the default display type.

Qualifier

TO file-spec
Specifies the output file. If none is specified, SYS$OUTPUT is the default.

Example

```
NCP>SHOW KNOWN NODES
```

This command displays summary information for all known nodes.

TELL

Use the TELL prefix to identify the executor node for a single NCP command.

This command is used when the load host is the executor.

Format

TELL node-component parameter [...] command-line

Node Component

node-spec
Specifies a node name or address optionally followed by access control information as specified for OpenVMS. Use one of the following forms:

node-id
node-id"username password"

The node-spec could be a logical name. It is possible to override access control in a logical name with explicit access control information in the command.

When using the Gateway, the node-spec identifies the Gateway node. The command line is therefore executed at the Gateway node.

Command Parameters

PASSWORD *password*
Identifies the password for access control verification at the designated executor node.

USER *user-id*
Identifies the username for access control verification at the designated executor node.

DESCRIPTION

The TELL command sets the executor for only one command and must prefix the command for which it is intended.

This command allows you optionally to specify access control information in one of two ways, either as part of the *node-spec* or as distinct parameters.

Examples

1. `NCP>TELL TRNTO"GRAY MARY" SHOW KNOWN LINES`

   This command uses access control information to set the executor to node TRNTO where the SHOW KNOWN LINES command executes. The information is displayed locally.

2. `NCP>TELL TRNTO USER SYSTEM PASSWORD MANAGER SHOW KNOWN LINES`

   This command uses an alternate access control format to have node TRNTO display known lines.

TRIGGER NODE

The TRIGGER NODE command triggers the bootstrap mechanism of a target node to allow the node to load itself. Use this command to initiate the loading sequence for an unattended system. This command should be executed on the load host, not on the Gateway.

This command is used when the load host is the executor.

Format

TRIGGER node-component parameter [...]

Node Component

NODE *node-id*
Identifies the node whose bootstrap is to be triggered.

Command Parameters

PHYSICAL ADDRESS *E-address*
Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node currently uses to identify itself. The value is the Ethernet physical address the target node has set for itself, or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter if associated with the target node in the executor node's volatile database.

SERVICE PASSWORD *password*
Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF.

VIA *circuit-id*
Identifies the circuit over which the operation is to take place.

Examples

1. `NCP>TRIGGER NODE NYC`

   This command triggers the bootstrap mechanism on node NYC to initiate a load operation.

2. `NCP>TRIGGER NODE NYC SERVICE PASSWORD FFFFFFFFFFFFFF VIA UNA-2`

   This command provides a service password in order to trigger the bootstrap mechanism on node NYC and to initiate a load operation over circuit UNA-2.

TRIGGER VIA

The TRIGGER VIA command triggers the bootstrap mechanism of a target node using the specified circuit to allow the node to load itself. Use this command to initiate the loading sequence for an unattended system. This command should be executed on the load host, not on the Gateway.

This command is used when the load host is the executor.

Format

TRIGGER VIA circuit-id parameter [...]

Command Parameters

PHYSICAL ADDRESS *E-address*
Applies only to nodes on Ethernet circuits. Specifies the Ethernet address that the target node currently uses to identify itself. The value is the Ethernet physical address the target node has set for itself, or, if the target node has not set an Ethernet address, the HARDWARE ADDRESS parameter if associated with the target node in the executor node's volatile database. This parameter must be included if an Ethernet circuit is specified in VIA circuit-id.

SERVICE PASSWORD *password*
Identifies the password required to trigger the bootstrap mechanism on the target node. The password is a hexadecimal number in the range 0 to FFFFFFFFFFFFFFFF. The word SERVICE is optional.

Examples

1. `NCP>TRIGGER VIA UNA-0 PHYSICAL ADDRESS AA-00-04-00-07-04`

   This command triggers the bootstrap mechanism at the target node whose Ethernet physical address on circuit UNA-0 is specified in the command.

ZERO CIRCUIT

The ZERO CIRCUIT command resets the counters for the circuit(s).

You use this command when the Gateway is the executor.

Format

ZERO circuit-component *[COUNTERS]*

Circuit Components

CIRCUIT *circuit-id*
Resets counters for the specified circuit.

KNOWN CIRCUITS
Resets counters for all known circuits.

Example

```
NCP>ZERO KNOWN CIRCUITS
```

This command resets the counters for all known circuits.

ZERO LINE

The ZERO LINE command sets line counters to zero.

You use this command when the Gateway is the executor.

Format

ZERO line-component *[COUNTERS]*

Line Components

ACTIVE LINES
Indicates that information for all active lines is to be displayed.

KNOWN LINES
Indicates that information for all known lines is to be displayed.

LINE *line-id*
Indicates that information for the specified line is to be displayed.

Example

```
NCP>ZERO KNOWN LINES COUNTERS
```

This command resets all line counters for all known lines.

ZERO NODE

The ZERO NODE command resets the node counters on the local node.

You use this command when the Gateway is the executor.

Format

ZERO node-component *[COUNTERS]*

Node Components

ACTIVE NODES
Resets the node counters for all reachable (active) nodes.

KNOWN NODES
Resets the counters for all known nodes.

NODE *node-id*
Resets the counters for the specified node.

Example

```
NCP>ZERO ACTIVE NODES
```

This command resets the counters for all reachable nodes.

A
EBCDIC/DMCS Translation Tables

The IBM system uses the Extended Binary Coded Decimal Interchange Code (EBCDIC) character set. Digital Equipment Corporation uses the DEC Multinational Character Set (DMCS). DMCS is a superset of the standard ASCII character set.

Some DECnet SNA access routines (for example, 3270 TE for OpenVMS) use translation tables to translate one character set into another.

You can modify the translation tables with the following two files provided in the management portion of the DECnet SNA OpenVMS Gateway distribution kit.

- A prefix file called SNATRAPRE.MAR
- A translation table template called SNATRATBL.MAR

The prefix file uses the ANSI Standard X3.26 1970 EBCDIC-to-ASCII translation tables. The translation template is the file you edit to modify the translation tables. You can change the tables on any node that is running the Gateway management software.

A.1 Macros for Modifying the Translation Tables

If the standard translation tables do not suit your needs, you can modify them by specifying macros in the file SNATRATBL.MAR. This file is provided on your Gateway distribution kit. You should copy it from SYS$LIBRARY into your current default directory. You can edit SNATRATBL.MAR with any editor supported by your system.

Use the macros described to make any changes you need in the translation tables. There are three macros you can specify. The arguments for all three macros are:

eb The EBCDIC code for the character you want to translate.

as The DMCS code for the character you are translating to. (If you want to specify the actual DMCS display character, instead of just the code, you can. To do this, enter a single quotation mark before you type the character; for example, ′!, ′A, ′g, and so on.)

The macros are:

1. EB2AS *eb, as*

   The EB2AS macro lets you change an entry in the EBCDIC-to-DMCS table without affecting the DMCS-to-EBCDIC table.

   For example:

   EB2AS 5A, ′!

   In this example, the EBCDIC hexadecimal code 5A is translated to the DMCS exclamation point (hexadecimal code 21). The macro does not affect the translation of a DMCS exclamation point to its EBCDIC equivalent.

2. AS2EB *as, eb*

   The AS2EB macro lets you change an entry in the DMCS-to-EBCDIC table without affecting the EBCDIC-DMCS table.

   For example:

   AS2EB ′[, 5F

   In this example, the DMCS open bracket character (hexadecimal code 5B) is translated to the EBCDIC hexadecimal code 5F. The macro does not affect the translation of the EBCDIC code 5F to DMCS.

3. REVTRA *eb, as*

   The REVTRA macro combines the functions of the EB2AS and AS2EB macros, and lets you change the same translation in both the DMCS-to-EBCDIC and EBCDIC-to-DMCS tables.

   For example:

   REVTRA 4A, A2

   In this example, the macro changes the EBCDIC-to-DMCS translation table so that the EBCDIC character represented by the hexadecimal code 4A translates to a DMCS cent sign (hexadecimal code A2.) The DMCS-to-EBCDIC translation table is also changed so that a DMCS cent sign translates to the EBCDIC character represented by the hexadecimal code 4A.

A.2 Building Translation Tables

Before you edit the template file, copy it from SYS$LIBRARY to your current default directory. By doing this, you save the original source file.

Edit the file using any editor your system supports. When you have changed the template file to your satisfaction, perform the following steps:

1. Use the appropriate command for your system, assemble the template file you just edited with the prefix file on your system called SNATRAPRE.MAR:

   For OpenVMS VAX:

   ```
   $ MACRO/OBJECT=SNATRATBL SYS$LIBRARY:SNATRAPRE+SYS$DISK:[]SNATRATBL
   ```

   For OpenVMS AXP:

   ```
   $ MACRO/MIGRATE/OBJECT=SNATRATBL SYS$LIBRARY:SNATRAPRE+SYS$DISK:[]SNATRATBL
   ```

   When you assemble the template file, you create an object file containing two 256-byte translation tables labeled $AS2EB:: and $EB2AS::. This object file can be linked into a user application program.

2. Link the template file to create the translation table:

   ```
   $ LINK/SYSTEM/HEADER SNATRATBL
   ```

3. Copy the image file to an accessible area. The name you specify for the translation table file depends on the access routine you are using. See the product-specific documentation for the default file name your access routine uses for its translation table.

   ```
   $ COPY SNATRATBL.EXE SYS$LIBRARY:access-routine-file-name.TBL
   ```

A.3 Examples

This section gives two examples of how to modify translation tables. Example A–1 shows how to translate the ASCII left bracket to the EBCDIC cent sign. Example A–2 shows how to modify the standard translation tables to the translation tables used by the DECnet SNA 3270 Terminal Emulator for OpenVMS.

The changes that are described modify a version of the ANSI standard X3.26 1970 EBCDIC-to-ASCII translation table. Table A–1 shows these modifications:

Table A–1 Modifications to Translation Tables

DMCS Character	Hexadecimal Code	EBCDIC Character	Hexadecimal Code
¢	A2	¢	4A
|	7C	|	4F
!	21	!	5A
¡[1]	A1	dashed vbar	6A
[	5B	See [2]	
]	5D	See [2]	

[1]The display of these characters is dependent on the type of terminal you have.

[2]These characters translate to the EBCDIC SUB character, which has an EBCDIC code of 63 decimal (3F hexadecimal). The DMCS contains 256 characters. The first 128 characters are the same as the standard ASCII character set. None of the remaining characters map to a printable EBCDIC character and therefore translate to the EBCDIC SUB character.

] The following code segment translates the ASCII left bracket, hexadecimal code 5B, to the EBCDIC cent sign, hexadecimal code 4A. The change causes the EBCDIC cent sign to be translated into the ASCII cent sign, hex A2. When the REVTRA macro is used, it leaves the ASCII left bracket unmapped, and a second macro, AS2EB, is used to map the ASCII left bracket to the EBCDIC SUB character, hex code 3F.

Example A–1 Example 1

```
DMFILL  = 26.           ; This argument causes all the EBCDIC
                        ; characters that normally map to an ASCII
                        ; backslash in the standard table to map
                        ; to an ASCII SUB character, code 26
                        ; decimal, 1A hexadecimal.

REVTRA  4A,A2           ; Map the EBCDIC cent character (4A)
                        ; to/from the ASCII cent character (A2).

AS2EB   5B,3F           ; Map the ASCII "[" (5B) to the EBCDIC
                        ; SUB character (3F).
```

(continued on next page)

Example A–1 (Cont.) Example 1

```
;
; The preceding macro could also be written in the following way:
;         AS2EB ’[,3F
```

This example shows the macros used to modify the standard translation tables to the translation tables used by the 3270 TE for OpenVMS. Table A–1 describes the required changes.

Example A–2 Example 2

```
DMFILL  = 26.

REVTRA  4A,A2           ; Map the EBCDIC cent character (4A)
                        ; to/from the ASCII cent character (A2).
                        ; Because this macro leaves ASCII "[" (5B)
                        ; still mapped to the EBCDIC cent character
                        ; (4A) , it must be remapped.

REVTRA  4F,7C           ; Map the EBCDIC "|" (4F) to/from
                        ; the ASCII "|" (7C).

REVTRA  6A,A1           ; Map EBCDIC "dashed vbar" (6A) to/from ASCII
                        ; inverted ! (A1).

REVTRA  5A,’!           ; Map EBCDIC "!" (5A) to/from ASCII "!" (21).

AS2EB   ’],3F           ; Map ASCII "]" (5D) to the EBCDIC SUB
                        ; character (3F).

AS2EB   5B,3F           ; Map the ASCII "[" (5B) to the EBCDIC
                        ; SUB character (3F).
```

B

Counter Summary

Counters are statistics on performance and errors that are kept for components such as lines, circuits, and so on. This information is useful alone, or in conjunction with logging information, in evaluating the performance of a component.

Each component has a timer parameter, COUNTER TIMER, that causes automatic logging of counters. When the counter timer expires, if event logging is enabled for information messages, counters are logged in an event message to the DECnet host and the counters are reset to zero.

You display counters on your screen with the SHOW command, and set them to zero with the ZERO command.

This appendix describes the counters for the following components:

- SDLC line
- Channel-attached line
- SDLC circuit
- Channel-attached circuit
- PU
- LU
- Server

The contents of all counters are reported in decimal numbers. Table B–1 lists these counters and their qualifiers. Section B.1 describes each counter.

Table B–1 Gateway Counters and Qualifiers

Component	Counters	Qualifiers
SDLC Line	Seconds since last zeroed	
	Bytes received	
	Bytes sent	
	Blocks received	
	Blocks sent	
	Carrier lost	
	Process errors	Transmit underrun
		Receive overrun
	Receive failures	Frame too long
		CRC error
	Send failures	Line turnaround error
Channel-attached Line	Seconds since last zeroed	
	Channel operations executed	
	Commands received	
	Bytes received	
	Bytes sent	
	Blocks received	
	Blocks sent	
	Buffers in use at peak	
	Attention buffers at peak	
	Process errors	Non-existent memory
		Device timeouts
		Zero function writes while active
		Function writes when not ready
	Channel errors	Invalid command byte
		Command byte parity error
		Bus-out parity error

(continued on next page)

Table B–1 (Cont.) Gateway Counters and Qualifiers

Component	Counters	Qualifiers
	Local buffer errors	Receive buffer ring empty
		Attention buffer unavailable
	Channel events	System resets
		Selective resets
		Halt-I/O executed
		Command to busy device
		Command to not-ready device
SDLC Circuit	Seconds since last zeroed	
	Bytes received	
	Bytes sent	
	Data blocks received	
	Data blocks sent	
	Data errors inbound, including:	Reject sent
		CRC inbound
		Frame too long
	Data errors outbound	Reject received
	Remote buffer errors	RNR received, buffer unavailable
	Local buffer errors	RNR sent, buffer unavailable
	Remote station errors	Poll/Final bit error
		FRMR sent, Unrecognized command
		FRMR sent, Frame should have no I-field
		FRMR sent, Invalid N(R) received
		FRMR sent, I-frame too long
Channel-attached Circuit	Seconds since last zeroed	
	Bytes received	

(continued on next page)

Table B–1 (Cont.) Gateway Counters and Qualifiers

Component	Counters	Qualifiers
	Bytes sent	
	Data blocks received	
	Data blocks sent	
	Process Errors	Data length error
		Connect rejected, invalid parameters
		Command received while not initialized
		Unrecognized command received
		Connect received while connected
		Read start old received
		Write start old received
	Attentions sent	
	Attentions sent on read	
	Write channel programs	
	Read channel programs	
PU	Seconds since last zeroed	
	Valid REQMS received	
	Invalid REQMS received	
	REQMS receive threshold exceeded	
	ACTPUs received, including:	ACTPUs (Cold)
		ACTPUs (ERP)
	Negative responses	Invalid LU id
		Invalid SSCP id
		PU not active
		FM/TS profile error
		Incomplete TH
		Incomplete RH

(continued on next page)

Table B–1 (Cont.) Gateway Counters and Qualifiers

Component	Counters	Qualifiers
		Invalid FID
		RU length error
		Segmentation error
		Segmented RU error
		No session
		Immediate request mode error
	DACTPUs received	
	Total ACTLUs received	
	Total DACTLUs received	
	Active LUs	
LU	Seconds since last zeroed	
	ACTLUs received, including:	ACTLUs (Cold)
		ACTLUs (ERP)
	BINDS received	
	BINDS accepted	
	BINDS rejected	LU inactive
		No server connected
		BINDs rejected by application
		BINDs rejected by Gateway
	DACTLUs received	
	Resource Errors	
	Session establishment failures	NSPE received
		Negative response to INITSELF received
		LU takedown
SERVERS		
SNA-ACCESS	Seconds since last zeroed	

(continued on next page)

Table B–1 (Cont.) Gateway Counters and Qualifiers

Component	Counters	Qualifiers
	Logical links active	
	Link requests accepted	
	Link requests rejected	
	Links aborted by server	
	Links aborted by DEC application	
	Active connect channels	
	Active LU–LU session channels	
	Active SSCP–LU session channels	
	Connect/Listen accepts	
	Connect/Listen rejects	
	NSPEs	
	INITSELF failures	
	Abnormal session terminations	
	Connect channel failures	
DHCF	Seconds since last zeroed	
	Logical links active	
	Total LUs in LU lists	
	Maximum acquired LUs	
	Current acquired LUs	
	DECnet connect failures	
	DECnet logical link failures	
	Session protocol errors	
	Internal software errors	
	BINDs rejected	
	Buffer allocation failures	

B.1 Counter and Qualifier Definitions

This section gives a brief explanation of the counters for each component.

B.1.1 SDLC Line Counters

Seconds since last zeroed

The number of seconds that have elapsed since the line counters were set to zero. This counter provides a time frame for other counter values.

Bytes received

The number of bytes of data received by the Gateway node over the specified line. This includes all the bytes from the Link Header (LH) to the Link Trailer (LT).

Bytes sent

The number of bytes of data sent by the Gateway-ST to the IBM system over the specified SDLC line.

Blocks received

The number of blocks of data received by the Gateway node over the specified line.

Blocks sent

The number of blocks of data sent by the Gateway-ST to the IBM system over the specified SDLC line.

Carrier lost

The number of times communications are lost because of a problem with the line.

Process errors

The number of errors that occurred while blocks of data were being processed by the Gateway node over the specified line. Qualifiers for this counter include:

- Transmit underrun—The device has not received a character as soon as it expected, so it sends the previous character repeatedly until it receives the next character.
- Receive overrun—A character received at the device could not be accepted by the device driver before the next character arrived. The first character was therefore overwritten in the device buffer. (Receive overruns are caused by excessive line speeds.)

Receive failures

The number of times that data was not received by the Gateway node over the specified line. Qualifiers for this counter include:

- Frame too long—The size of the received data block was greater than the size of the largest available buffer. If this happens often, it could indicate a configuration problem. If it happens occasionally, it could indicate a line error corrupting the end-of-message flag.
- CRC error—The Cyclic Redundancy Check (CRC) on data received did not match the value calculated by the sender.

Send failures

The number of errors that occurred while blocks of data were being sent by the Gateway-ST to the IBM system over the specified SDLC line. The qualifier for this counter is:

- Line turnaround error—An error occurred between receiving and sending data over the line.

B.1.2 Channel-Attached Line Counters

Seconds since last zeroed

The number of seconds that have elapsed since the counters were set to zero. This counter provides a time frame for other counter values.

Channel operations executed

The number of channel operations executed. A channel operation indicates either processing of a received channel command, or sending of asynchronous status to the channel.

Commands received

The number of channel commands received.

Bytes received

The number of bytes of data received by the Gateway node over the specified line.

Bytes sent

The number of bytes of data sent by the Gateway-CT to the IBM system over the specified channel-attached line.

Blocks received

The number of blocks of data received by the Gateway node over the specified channel-attached line.

Blocks sent

The number of blocks of data sent by the Gateway-CT to the IBM system over the specified channel-attached line.

Buffers in use at peak

This counter records the highest number of receive buffers that were in use simultaneously. This information can be useful in tuning the value of the RECEIVE BUFFERS parameter.

Attention buffers at peak

This counter records the highest number of attention buffers that were in use simultaneously. This information can be useful in tuning the value of the ATTENTION BUFFERS parameter.

Process errors

The number of errors that occurred because of a device driver or DEC ChannelServer malfunction. Qualifiers for this counter include:

- Non-existent memory — An attempt was made to reference non-existent memory. The problem could be a result of a software error, a memory error, a device error, or a bus error.
- Device timeouts — The device failed to interrupt within 10 seconds. Either the device or the Gateway software is malfunctioning. This can occur when the IBM system is being reinitialized. If it occurs at other times, either the device or the Gateway software is malfunctioning.
- Zero function writes while active — The Gateway software is not interacting correctly with the device.
- Function writes when not ready — The Gateway software is not interacting correctly with the device.

Channel errors

The number of errors that occur during the operation of the channel protocol. Qualifiers for this counter include:

- Invalid command byte — An unsupported channel command was received. This error could be the result of incorrect IBM parameters, an incorrect Gateway configuration, or a hardware error in the channel interface, channel, or channel cables.
- Command byte parity error — A hardware error caused a command byte to have incorrect parity.
- Bus-out parity error — A hardware error caused a data byte to have incorrect parity.

Local buffer errors

The number of errors that occur when the device driver runs out of buffers. Qualifiers for this counter include:

- Receive buffer ring empty — The driver received a Write command from the channel; however, the driver had no buffers available to process the command. This qualifier indicates the Gateway-CT's performance is suffering. If you notice this counter incremented frequently during normal operation, increase the RECEIVE BUFFERS parameter.
- Attention buffer unavailable — The driver ran out of the buffers used to signal internal events. This qualifier indicates a problem with the Gateway software.

Channel events

This counter is used by the Gateway's system administrator to note activity on the channel. Qualifiers for this counter include:

- System resets — The IBM system executed a system reset on the channel. A system reset occurs when the IBM system is rebooted or the channel fails.
- Selective resets — The IBM system resets the channel address. If this qualifier occurs frequently there might be a problem with the Gateway software or an error in the configuration. Selective resets can also occur when the IBM system is initializing.
- Halt-I/O executed — The IBM channel program halted an I/O operation. If this qualifier occurs frequently there might be a problem with the Gateway software or an error in the configuration.
- Command to busy device — The channel sent a command to an address that was already busy.
- Command to not-ready device — The channel sent a command to an address that was not ready. This qualifier should not appear during normal operation.

B.1.3 SDLC Circuit Counters

Seconds since last zeroed

The number of seconds that have passed since the circuit counters were set to zero. All other counter values are set within the time indicated by this counter.

Bytes received

The total number of bytes in all information frames (I-frames) received by the Gateway node over the specified SDLC circuit. This includes the Request Header (RH) and the Request Unit (RU).

Bytes sent

The total number of bytes in all information frames (I-frames) sent by the Gateway-ST to the IBM system over the specified SDLC circuit.

Data blocks received

The number of information frames (I-frames) received by the Gateway node over the specified SDLC circuit.

Data blocks sent

The number of information frames (I-frames) sent by the Gateway-ST to the IBM system over the specified SDLC circuit.

Data errors inbound, including:

The number of errors that occurred while blocks of data were being received by the Gateway node over the specified circuit. The qualifiers for this counter are:

- Reject sent— An SDLC REJ message was sent to the IBM system because of an SDLC level error.
- CRC inbound–The Gateway-ST received a frame with a CRC error.
- Frame too long—The IBM host sent a frame that was too big for the buffers. Increase the buffer size by altering the BUFFER SIZE parameter for the line.

Data errors outbound

The number of errors that occurred while blocks of data were being sent to the IBM system over the specified circuit. The qualifier for this counter is:

- Reject received—An SDLC REJ message was received by the Gateway node.

Remote buffer errors

The number of times the IBM system did not have adequate buffer space to receive data from the Gateway node. The qualifier for this counter is:

- RNR received, buffer unavailable—The number of Receive Not Ready frames received by the Gateway node. The RNR frames were caused by the IBM system being temporarily unable to accept any more data from the Gateway node because the IBM 372*x* had insufficient buffer space.

Local buffer errors

The number of times the Gateway-ST did not have adequate buffer space to receive data from the IBM system. The qualifier for this counter is:

- RNR sent, buffer unavailable—The number of Receive Not Ready frames sent by the Gateway node. The RNR frames were caused by the Gateway-ST being temporarily unable to accept any more data from the IBM system because it had insufficient buffer space.

Remote station errors

The number of SDLC protocol errors detected by the Gateway-ST. Qualifiers for this counter include:

- Poll/Final bit error—The Gateway-ST received a poll bit when it had already received one.
- FRMR sent, Unrecognized command—FRMR (Frame Reject) sent because the control byte of the frame did not correspond to any supported SDLC frame or command type.
- FRMR sent, Frame should have no I-field—The Gateway-ST received an invalid frame that contained an I-field when the control field indicated there should not be an I-field.
- FRMR sent, invalid N(R) received—An Information or Supervisor frame was received with an invalid receive-sequence number - N(R).
- FRMR sent, I-frame too long—The IBM host sent a frame that was too big for the buffers. Increase the buffer size by altering the BUFFER SIZE parameter for the line.

Local station errors

Not currently used.

Test frames received

The number of TEST frames without an error received by the Gateway node over the specified circuit.

Test frames received, bad CRC

The number of TEST frames with a Cyclic Redundancy Check (CRC) error received by the Gateway node over the specified circuit.

Polls received

The number of times the IBM system has polled the Gateway node over the specified circuit. This counter increases regularly as long as the Gateway-ST is communicating with the IBM system.

Frame retransmits

The number of times the IBM host requested the Gateway-ST to retransmit a frame.

B.1.4 Channel-Attached Circuit Counters

Seconds since last zeroed

The number of seconds that have passed since the circuit counters were set to zero. All other counter values are set within the time indicated by this counter.

Bytes received

The total number of bytes in all write commands received by the Gateway node over the specified channel-attached circuit. This includes the Request Header (RH), the Request Unit (RU), and the Transmission Header (TH).

Bytes sent

The total number of bytes in all read commands sent by the Gateway-CT to the IBM system over the specified channel-attached circuit. This includes the Request Header (RH), the Request Unit (RU), and the Transmission Header (TH).

Data blocks received

The number of write commands received by the Gateway node over the specified channel-attached circuit.

Data blocks sent

The number of read commands sent by the Gateway-CT to the IBM system over the specified channel-attached circuit.

Process errors

The number of errors that occurred in the operation of the channel protocol. Qualifiers for this counter include:

- Data length error — The size of the data buffer received from the channel does not match the data buffer size contained in the Link Header (LH).
- Connect rejected, invalid parameters — The Connect command received from the channel had invalid parameters. This error occurs when the size of the host buffers multiplied by the number of the host buffers is not large enough to contain a Link Header (LH) (4-32 bytes), a Transmission Header (TH) (6 bytes), a Request Header (RH) (3 bytes), and at least 256 bytes of RU data. This error can also occur if the size of the host buffer is an odd number of bytes, if the size of the host buffer is less than 78 bytes, or if the Link Header size is odd or greater than 32 bytes.

- Command received while not initialized — A channel command other than a Sense, Sense-ID, or Control was received when the Gateway-CT was not initialized.
- Unrecognized command received — An unsupported command was received by the Gateway-CT.
- Connect received while connected — The Gateway-CT received an initializing command when it was already initialized.
- Read start old received — The host requested retransmittal of data.
- Write start old received — The host retransmitted data to the Gateway-CT.

Attentions sent

The number of times the Gateway-CT presented status indicating that it had data to transmit to the host, including Attentions sent on read.

Attentions sent on read

The number of times the Gateway-CT presents an ending status to a Read command indicating it has data to send to the host.

Write channel programs

The number of Write channel programs executed by the host.

Read channel programs

The number of Read channel programs executed by the host.

B.1.5 PU Counters

Seconds since last zeroed

The number of seconds that have passed since the PU counters were set to zero. All other counter values are set within the time indicated by this counter.

Valid REQMS received

The number of valid SNA REQMS (Request Maintenance Statistics) messages that were received by the Gateway node. REQMS messages are valid when the request for data is supported and the REQMS threshold has not been exceeded.

Invalid REQMS received

The number of invalid SNA REQMS (Request Maintenance Statistics) messages that were received by the Gateway node. REQMS messages are invalid when either the request for data is not supported, or the REQMS threshold has been exceeded.

REQMS receive threshold exceeded

The number of times SNA REQMS messages have been received at shorter time intervals than those specified by the PU QUERY THRESHOLD parameter.

The number of SNA ACTPU messages received by the Gateway node.

- ACTPUs (Cold)—The number of cold ACTPUs received. Cold ACTPUs have the COLD type of activation in which the SSCP-PU session (and all other sessions involving Network Addressable Units (NAUs) on this node) is reset before it is activated.

  If the PU is already active, any SSCP–LU or LU–LU sessions associated with this PU are terminated.

- ACTPUs (ERP)—The number of warm ACTPUs received. Warm ACTPUs have the ERP (Error Recovery Procedure) type of activation in which the SSCP-PU session is established without disrupting the parameters related to that session (or other sessions involving NAUs on this node).

  If the PU is already active, any associated SSCP–PU and LU–LU sessions are left untouched, and the SSCP–PU session is not reset.

Negative responses

The number of negative responses, including the following:

- Invalid LU id—An SNA ACTLU, DACTLU, BIND, or UNBIND request was received for an LU that was not defined for this PU.
- Invalid SSCP id in ACTPU request—The SSCP ID supplied in the ACTPU request did not match any of those specified with the SET PU SSCP ID command. Note: if you do not set this parameter, the Gateway does not check the SSCP IDs.
- PU not active—An SNA DACTPU, ACTLU, DACTLU, BIND, or UNBIND request was received, but the PU has not been activated.
- FM/TS profile error—The IBM host tried to start an SSCP–PU session with an FM or TS profile that is not supported by the Gateway.
- Incomplete TH—The transmission header was not complete.
- Incomplete RH—The request/response header was not complete.
- Invalid FID—The type format identifier is not supported by the Gateway.
- RU length error—The request/response unit was too long.
- Segmentation error—Path control detected a segmentation problem.

- Segmented RU error—Path control detected a segmented request/response unit problem.
- No session—The IBM host tried to send the Gateway some data without starting a session.
- Immediate request mode error—The IBM host sent a request before receiving a response to the previous request.

DACTPUs received

The number of SNA DACTPU messages received by the Gateway for the specified PU.

Total ACTLUs received

The total number of ACTLUs received.

Total DACTLUs received

The total number of DACTLUs received.

Active LUs

The number of LUs on this PU that are in an active session with an SSCP.

B.1.6 LU Counters

The following counters are for the current session and all sessions since the LU was defined or the counters were zeroed. In other words, these counters are cumulative.

Seconds since last zeroed

The number of seconds that have elapsed since the LU counters were set to zero.

ACTLUs received, including:

The number of SNA ACTLU messages received by the Gateway for the specified LU.

- ACTLUs (Cold)—The number of cold ACTLUs received. When an ACTLU (cold) is received and the LU is already active, any LU–LU session associated with this LU is terminated.
- ACTLUs (ERP)—The number of warm ACTLUs received. When an ACTLU (ERP) is received and the LU is already active, any associated LU–LU session is left untouched and the SSCP–LU session is not reset.

BINDs received

The number of BIND messages received by the Gateway node from the IBM system.

BINDs accepted

The number of BIND messages accepted by the Gateway node.

BINDs rejected

The number of BIND messages rejected by the Gateway node.

- LU inactive—No SNA ACTLU request received yet for that LU.
- No server connected—No server connected for that LU. A server must be connected to receive the BIND message.
- BINDs rejected by application—The access routine rejected the BIND because it was unsuitable for the type of session the access routine required.
- BINDs rejected by Gateway—The Gateway rejected the BIND because it contained an FM/TS profile not supported by the Gateway.

DACTLUs received

The number of SNA DACTLU messages received by the Gateway.

Resource errors

The number of times attempts to allocate resources have failed. A large number indicates an overloaded system.

Session establishment failures

A server has connected and the Gateway has sent an INITSELF to start an LU–LU session, but a BIND is not returned by the IBM host for the following reasons:

- NSPE received—A network services procedure error occurred because the IBM services requested are not available.
- Negative response to INITSELF received—Occurs if, for example, the application you have requested does not exist.
- LU takedown—The LU was inactivated while the session was being established.

The rest of the LU counters are for the latest session: this is either the current session or, if there is no LU–LU session active or a BIND message has not been received, the most recent session.

B.1.7 Server Counters

The server counters are server specific. This section explains DHCF and SNA-ACCESS counters. For a list of RJE counters, see the *DECnet SNA Remote Job Entry for OpenVMS User's and Operator's Guide*.

B.1.7.1 SNA-ACCESS Server Counters

Seconds since last zeroed

The number of seconds that have passed since the server counters were set to zero. All other counter values are set within the time indicated by this counter.

Logical links active

The number of DECnet logical links that are active.

Link requests accepted

The total number of DECnet logical links that have been, or are, active.

Link requests rejected

The total number of DECnet logical link requests that have been rejected. A request for a link would be rejected, for example, if it would cause the number of links to exceed the maximum specified by the MAXIMUM LINKS parameter.

Links aborted by server

The number of DECnet logical links the server terminated.

Links aborted by DEC application

The number of DECnet logical links terminated by the application interface. For example, a normal exit from an access routine using Ctrl/Z.

Active connect channels

The number of active channels to LU services.

Active LU–LU session channels

The number of active LU sessions.

Active SSCP–LU session channels

The number of active SSCP–LU sessions.

Connect/Listen accepts

The number of times the access control information has been accepted.

Connect/Listen rejects

The number of times the access control information has been rejected.

NSPEs

The Network Services Procedure Errors used by SSCP to indicate that the initialization failed after a positive response was sent.

INITSELF failures

This counter indicates the number of times the request to SSCP to initialize the session failed.

Abnormal session terminations

This indicates abnormal line activity.

Connect channel failures

This indicates a breakdown in communications between the server and LU services.

B.1.7.2 DHCF Server Counters

Seconds since last zeroed

The number of seconds that have passed since the server counters were set to zero. All other counter values are set within the time indicated by this counter.

Logical links active

This is the number of current DECnet logical links.

Total LUs in LU list

The total number of LUs in the DHCF LU List. The names of the LUs in the list are given in the DHCF characteristics display.

Maximum acquired LUs

This is the highest number of current acquired LUs. If this number is consistently smaller than the number of LUs in LU List, the DHCF LU List should be reduced.

Current acquired LUs

This is the number of LUs in use. Current acquired LUs equals the total of acquired and logged-on LUs.

The following counters should be zero. If they are not, check the event log. The buffer allocation failures are not logged because event logging uses buffers.

- DECnet connect failures
- DECnet logical link failures
- Session protocol errors

- Internal software errors
- BINDs rejected
- Buffer allocation failures

B.2 IBM RECFMS Messages

Both Gateways can send RECFMS (Record Formatted Maintenance Statistics) counters to the IBM system. These messages contain counters requested by the IBM Network Problem Determination Application (NPDA), using REQMS (Request Maintenance Statistics) messages. NPDA is a problem determination program. It collects information about the IBM network, collates it in a database, and displays it at the IBM system. Network operators on the IBM system can use this program to get information about the Gateway.

Use the PU parameter QUERY THRESHOLD to limit the frequency with which the Gateway handles REQMS messages. Refer to the command description of SET PU in Chapter 13 for information on QUERY THRESHOLD. The REQMS messages are not affected by the SNANCP ZERO COUNTERS commands.

The Gateway-ST supports the following RECFMS counters:

- Seconds since last zeroed
- Total transmitted I-frames
- Total received I-frames
- Write retries
- FCS errors
- DCE errors
- Nonproductive timeouts
- SDLC command rejects

The Gateway-CT supports the following RECFMS counters:

- Not initialized-control
- Command reject
- Not initialized-sense
- CA Bus out parity-select
- CA Bus out parity-write

- CA Internal parity-read
- Data check
- Data length check-received
- Connect received
- Connect parameter error
- Incorrect sequence

For more information on NPDA, see the IBM *Network Problem Determination Application User Reference.*

C

Gateway Management Utility Messages

This appendix lists messages you may encounter while using the following Gateway management utilities:

- SNANCP
- SNAP
- NETTRACE
- SNATRACE

Gateway management utility messages have the following format:

FACILITY-L-IDENT, TEXT

where

FACILITY	Is the name of the facility or program that generates the message
L	Is a severity level indicator with one of the following values:

Code	Meaning
S	Success
I	Information
W	Warning
E	Error
F	Fatal, or severe error

IDENT	Is the identification of the message
TEXT	Is the explanation of the message

The following is a sample message:

```
%SNANCP-F-AMBCMD, ambiguous command
```

In this appendix severity levels are omitted. Messages are listed in alphabetical order according to their IDENT.

ABNSESTER, session terminated abnormally

Facility: SNATRACE

Explanation: The session between the IBM system and the Gateway system might have terminated for one of the following reasons:

- The hardware communications link failed.
- The IBM system deactivated the physical unit (PU).
- The IBM system deactivated the line leading to your system.

User Action: Determine why the link was lost. Try again when the connection to IBM returns.

AMBCMD, ambiguous command *keyword*

Facility: SNANCP

Explanation: You did not specify enough characters of a keyword to distinguish it from other keywords that could be used in this context.

User Action: Reenter the command. Use a minimum of four characters to make a keyword unique.

ASSIGN, failed to assign channel(s)

Facility: SNATRACE

Explanation: SNATRACE could not assign a channel for the trace request you specified. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

Can not assign channel to *device-name*,

Facility: SNAP

Explanation: SNAP cannot assign a channel to a terminal to read user input commands. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

BADFILE, invalid trace input file format

Facility: SNATRACE

Explanation: An error was detected in data read from a trace input file.

User Action: Make sure that the file you specified as an input file is a trace data input file.

BADKEY, *keyword* is an invalid keyword

Facility: SNATRACE

Explanation: You specified an invalid keyword. The message displays the rejected portion of the command.

User Action: Check the command for a spelling or syntax error. Refer to Chapter 12 for valid qualifiers, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

BADLOGIC, internal logic error (module *name*, routine *name*)

Facility: SNATRACE

Explanation: An internal error was detected in the SNATRACE program.

User Action: Please record the information displayed on your screen and report the error to your Gateway manager. Contact your Digital service representative.

BADMSG, invalid management message, *message*

Facility: SNANCP

Explanation: The form of the data from the Gateway Network Management Listener (SNANML) could not be parsed correctly. This is an internal software error. The reason for it is specified at the end of the message.

User Action: Report this problem and the circumstances that caused it to your Digital service representative.

BADPURGE, failed to purge output files

Facility: SNATRACE

Explanation: SNATRACE did not purge versions of the trace output files specified with the /VERSION_LIMIT qualifier. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

BADVALUE, *value* is an invalid keyword value

Facility: SNATRACE

Explanation: You specified a value for a command that is outside of the valid range. This error can also be caused by a typographical error.

User Action: Check the range of values in Chapter 12, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

BEGIN, beginning trace at *date* and *time*

Facility: SNATRACE

Explanation: This message tells you when the trace started.

User Action: None.

BUFSIZE, maximum buffer size exceeded

Facility: SNATRACE

Explanation: The data size specified with the /SIZE qualifier exceeds the size of the buffer available to contain trace data.

User Action: Decrease the value specified with the /SIZE qualifier and reenter the SNATRACE command.

Can not connect to server on node *gateway-node-name*,

Facility: SNAP

Explanation: This message indicates that the information you entered is not recognized by the system. There might be a syntax error, an invalid command, or a node name unknown to the system. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

CANCEL, command cancelled

Facility: SNATRACE

Explanation: You issued a Ctrl/Z to cancel a command while SNATRACE was processing a command. The command is cancelled; you are returned to the SNATRACE prompt.

User Action: Reenter the command.

CHANSTRACED, *n* channels are currently being traced

CHANTRACED, 1 channel is currently being traced

Facility: NETTRACE

Explanation: Tracing has been started and *n* channels are currently being traced. This is the number of lines or channels that match the tracepoint specified. However, if new channels are created, they will also be traced.

User Action: Use SHOW /FULL to obtain a list of channels being traced.

CLOSEIN, error closing *file-spec* as input

Facility: SNATRACE

Explanation: The specified file could not be closed. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

CLOSEOUT, error closing *file-spec* as output

Facility: SNATRACE

Explanation: The specified file could not be closed. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

CMDCAN, command cancelled

Facility: SNANCP

Explanation: You typed Ctrl/Z in response to a prompt. Ctrl/Z cancels the current command and returns to the SNANCP> prompt.

User Action: If you cancelled the command by mistake, reenter your original command at the prompt.

CMDERR, I/O error reading commands

Facility: SNATRACE

Explanation: RMS returned an error while reading commands from SYS$INPUT. The RMS error is displayed for additional information.

User Action: For information on the RMS error message, refer to the *OpenVMS System Messages and Recovery Procedures*.

COMMAND, error getting command line

Facility: SNAP

Explanation: The input line is too long. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

CONFAIL, failed to connect to trace collector

Facility: NETTRACE

Explanation: TRACE failed to connect to the trace collector object.

User Action: Take corrective action based on the accompanying messages.

CONFQUAL, conflicting qualifiers

Facility: SNATRACE

Explanation: You used qualifiers that cannot be used together in the same command. You would receive this error message if you entered TRACE/PU/SESSION, for example.

User Action: Decide which qualifiers you want to use and reenter the command.

CONNEC, unable to connect to listener on node *node-id*

Facility: SNANCP

Explanation: SNANCP is unable to connect to the Gateway Network Management Listener (SNANML) to perform network operations. The secondary reason is displayed with this error for additional information. This error might also indicate that the network is not configured correctly or that all files are not present. (For example, you see this error if you do not specify the USE SYSTEM command before you enter another command.)

User Action: Refer to Chapter 13 to make sure you have correctly specified the USE SYSTEM command.

CONNECT, network connect failed

Facility: SNATRACE

Explanation: SNATRACE was unable to connect to the trace server program.

User Action: Enter an NCP SHOW NODE command to see whether the node is reachable and try again.

CONNECTING, connecting to trace collector ...

Facility: NETTRACE

Explanation: Your process is trying to connect to the trace collector.

User Action: None.

CREATEFILE, file *filename* has been created

Facility: NETTRACE

Explanation: A new output file has been created.

User Action: None.

DATALOST, trace data lost

Facility: SNATRACE

Explanation: The data transfer rate is faster than the data collection and recording rate.

User Action: Increase the number of available buffers with the /BUFFERS qualifier; then redo the trace operation.

DEASSIGN, error deassigning *device-name*

Facility: SNAP

Explanation: SNAP cannot disconnect the channel from the terminal. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

DETSTART, detached trace collector *process-name* has been started

Facility: NETTRACE

Explanation: A detached trace collector process has been started.

User Action: None.

DISCON, error disconnecting link to listener

Facility: SNANCP

Explanation: An error occurred while disconnecting the link to SNANML. The secondary message provides more information.

User Action: Take corrective action based on the information provided. If necessary, refer to the *OpenVMS System Messages and Recovery Procedures*.

DISCONN, error disconnecting from *gateway-node-name*

Facility: SNAP

Explanation: An error occurred while attempting to disconnect the logical link from the Gateway node. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

ENDED, trace ended at *date* and *time*

Facility: SNATRACE

Explanation: This message tells you when the trace ended.

User Action: None.

END_SAVED, end of saved trace records

Facility: NETTRACE

Explanation: If you have been using the BACK and NEXT commands, this message is displayed when you finish reexamining saved trace records and resume looking at new records.

User Action: None.

EOFILE, end of input reached

Facility: NETTRACE

Explanation: No more trace records are in the file being analyzed.

User Action: None.

ERRCRMBX, error creating mailbox *name*

Facility: NETTRACE

Explanation: An error occurred while the process was creating the mailbox that TRACE uses to communicate with the detached trace collector *name*.

User Action: Take corrective action based on the accompanying messages.

ERRDETPROC, error in creating detached process *name*

Facility: NETTRACE

Explanation: An error occurred while the process was creating the detached trace collector *name*. An accompanying message gives the reason for the failure.

User Action: Take corrective action based on the accompanying messages.

ERR_FIND_SYMBOL, error finding *symbol*

Facility: NETTRACE

Explanation: An error occurred when the process tried to activate a trace analysis routine. The symbol could not be found in the sharable image.

User Action: Contact your Digital service representative.

ERROR, unexpected error

Facility: NETTRACE

Explanation: An unexpected error has occurred.

User Action: Take corrective action based on the accompanying messages.

ERRSTART, error starting trace

Facility: NETTRACE

Explanation: Tracing could not be started.

User Action: Take corrective action based on the accompanying messages.

ERRSTOP, error stopping detached process

Facility: NETTRACE

Explanation: An error occurred while you were stopping a detached trace collector process.

User Action: Take corrective action based on the accompanying messages.

EXENTRIES, maximum output file entries exceeded

Facility: SNATRACE

Explanation: You have exceeded the maximum number of entries per output file. The allowable range is 2—32767.

User Action: If you specified this parameter, verify that it is within the allowable range. Take corrective action based on the accompanying messages.

EXSIZE, maximum output file size exceeded

Facility: SNATRACE

Explanation: You have exceeded the maximum size for output files.

User Action: Try using either the /ENTRIES= or /VERSION_LIMIT= qualifiers when running trace. Take corrective action based on the accompanying messages.

FATINTERR, fatal internal error

Facility: SNATRACE

Explanation: A program error occurred within the SNATRACE utility. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages. Report the problem to your Digital service representative.

FIELDLIM, number of fields in range exceeds limit

Facility: SNANCP

Explanation: A PU SSCP list can have only eight fields. The fields must be separated by commas.

User Action: Correct the list to include the proper number of comma-separated fields.

FILEANALYZED, file is already being analyzed

Facility: NETTRACE

Explanation: The file you specified in an ANALYZE command is already being analyzed by another user or process.

User Action: Specify another file name, or wait for the user or process to finish analyzing the file.

FOUNDTRPT, *tracepoint-name* tracepoint found - will be ignored

Facility: NETTRACE

Explanation: A tracepoint has been found but will not be analyzed because another tracepoint type has already been found.

User Action: Use ANALYZE/TRACE_LEVEL=*protocol_identifier* to analyze this tracepoint.

GATINTERR, internal error in Gateway node, code *code-number*, subcode *subcode-number*

Facility: SNATRACE

Explanation: An internal error occurred.

User Action: Contact your Digital service representative. Give the exact sequence of events up to the point when the error occurred, or give the information to your system manager when you report the problem.

HELP, error writing help text

Facility: SNATRACE

Explanation: An error occurred when writing the HELP text to SYS$OUTPUT. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

INCVER, incompatible version numbers

Facility: SNATRACE

Explanation: The version of SNATRACE you are using is not compatible with the trace server.

User Action: Ensure that your SNATRACE image is compatible with your version of the Gateway.

INIT, initialization failure

Facility: SNATRACE

Explanation: There was a resource failure in the SNATRACE program. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

INITERROR, error occurred during initialization

Facility: NETTRACE

Explanation: An unexpected error occurred during initialization.

User Action: Take corrective action based on the accompanying messages.

INPUT, terminal input error

Facility: SNAP

Explanation: SNAP cannot read the user input. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

INSFQUO, *quota-name* quota must be at least *n*

Facility: NETTRACE

Explanation: The current value for the user's quota *quota-name* is too small for NETTRACE to run.

User Action: Use AUTHORIZE to increase the quota to at least *n*.

INSGATRES, insufficient Gateway resources for session establishment

Facility: SNATRACE

Explanation: There are not enough resources to establish a trace session. Active sessions are currently using all available resources.

User Action: Wait until some of the sessions have finished; then try again.

INTERRHOS, server detected trace protocol error

Facility: SNATRACE

Explanation: The server detected a trace protocol error.

User Action: Contact your Digital service representative.

INTMSG, unexpected interrupt message received from server

Facility: SNATRACE

Explanation: The server sent an unexpected interrupt message.

User Action: Take corrective action based on accompanying messages.

INVACC, invalid access control specified for executor

Facility: SNANCP

Explanation: You specified invalid or conflicting access control information when you issued the USE SYSTEM command.

User Action: See Chapter 13 to verify the format of the USE SYSTEM command.

INVCIRCID, invalid circuit specification

Facility: SNATRACE

Explanation: You specified a *circuit-id* incorrectly. When using an SDLC circuit level trace, the *circuit-id* format for SNATRACE is SDLC-*nnn*, where *nnn* identifies a particular SDLC circuit. When using a channel-attached circuit level trace, the *circuit-id* format for SNATRACE is CHAN-*nnn*, where *nnn* identifies a particular channel-attached circuit.

User Action: Enter the correct *circuit-id*.

INVCMD, unrecognized command *command*

Facility: SNANCP

Explanation: SNANCP did not recognize the command you entered.

User Action: Check the appropriate chapter in this manual to verify the command and its format. Then reenter the command.

INVEVE, invalid event range *value*

Facility: SNANCP

Explanation: You supplied a range of event types that is not valid.

User Action: Reenter the valid event list.

INVFILE, invalid file specification *file-spec*

Facility: SNATRACE

Explanation: You have entered the file specification incorrectly. There is a spelling or syntax error. The invalid file specification is displayed in the message line.

User Action: Reenter the command with the correct file specification.

INVKEY, unrecognized keyword

Facility: SNANCP

Explanation: SNANCP did not recognize one of the keywords you entered in response to a prompt.

User Action: Check the appropriate chapter in this manual to verify the format of the command. Reenter the command. If you are responding to a prompt, choose one of the options displayed.

INVLU, invalid LU range *range*

Facility: SNANCP

Explanation: You specified a range of LU numbers that is not valid. A range of LU numbers must be consecutive (for example: 20–25). The numbers or ranges of numbers must be separated by a comma.

User Action: Reenter the command using a valid range of LU numbers.

INVLUID, invalid logical unit specification

Facility: SNATRACE

Explanation: You specified an incorrect *lu-id*. The *lu-id* format for SNATRACE is PU-id.*n*, where *n* identifies a particular session LU for this PU.

User Action: Enter the correct *lu-id*.

INVNODE, invalid node specification

Facility: SNATRACE

Explanation: The node specification does not conform to DECnet conventions.

User Action: Reenter the command with the correct node specification.

INVRANGE, invalid range *range*

Facility: NETTRACE

Explanation: *range* is not a valid range of values. A valid range is either a single value, a range of values, or *; for example, 4–5.

User Action: Reenter the command using a valid range.

INVREC, invalid record in file

Facility: NETTRACE

Explanation: The file being analyzed either is not a valid trace file or contains corrupted data.

User Action: Check the trace file name and analyze the file again.

INVRSP, invalid management response

Facility: SNANCP

Explanation: SNANCP could not recognize the form of the response from the Gateway Network Management Listener (SNANML), or the response was inappropriate.

User Action: Report this problem and the circumstances that caused it to your Digital service representative.

INVSHOW, invalid SHOW command

Facility: NETTRACE

Explanation: An error occurred parsing a SHOW command.

User Action: Take corrective action based on the accompanying messages.

INVSTART, invalid START command

Facility: NETTRACE

Explanation: An error occurred parsing a START command.

User Action: Take corrective action based on the accompanying messages.

INVTRPT, invalid tracepoint *tracepoint-name*

Facility: NETTRACE

Explanation: The syntax of *tracepoint-name* is invalid.

User Action: For a description of the valid tracepoint syntax, see Chapter 11.

INVVAL, unrecognized value

Facility: SNANCP

Explanation: SNANCP did not recognize one of the values you supplied in response to a prompt.

User Action: Check the appropriate chapter in this manual to verify the format of the command. Reenter the command.

INV_VALUE, invalid value *value*

Facility: NETTRACE

Explanation: The value specified is not a valid number.

User Action: Reenter the command using a valid number.

LIBFAOL, LIB$SYS_FAOL failed

Facility: SNATRACE

Explanation: Failed to format trace record.

User Action: Take corrective action based on accompanying messages.

LIBGETMSG, LIB$SYS_GETMSG failed

Facility: SNATRACE

Explanation: The system failed to get the message text associated with a MESSAGE-ID number.

User Action: Take corrective action based on the accompanying messages.

LIBGETVM, LIB$GET_VM failed

Facility: SNATRACE

Explanation: Failed to get virtual memory, insufficient memory.

User Action: Take corrective action based on the accompanying messages.

LINKERR, error on link to trace collector

Facility: NETTRACE

Explanation: The link to the trace collector failed.

User Action: Take corrective action based on the accompanying messages.

LIVEACTIVE, live tracing is already active

Facility: NETTRACE

Explanation: You attempted to start tracing while live tracing was already active.

User Action: None.

LOGIC, logic error in SNANCP, *error*

Facility: SNANCP

Explanation: A logic error occurred in SNANCP, which is an inconsistency caused by a program bug.

User Action: Report this problem and the circumstances that caused it to your Digital service representative.

LTCEXIT, trace collector *trace-collector-name* has exited

Facility: NETTRACE

Explanation: A trace collector exited. An OPCOM message tells why the process exited.

User Action: Take action based on the OPCOM message.

LTCINITERR, trace collector *name* failed to initialize

Facility: NETTRACE

Explanation: The detached trace collector *name* did not initialize correctly.

User Action: Contact your Digital service representative.

MAXPARM, maximum parameter count exceeded *string*

Facility: SNATRACE

Explanation: You entered a command incorrectly. This error might be caused by one of the following:

- Leaving blanks on a command line where a special character (for example a comma or plus sign) is required.
- Using symbol names or logical names which, when substituted or translated, contain embedded blank characters.
- Failing to place quotation marks around a character string with embedded blanks.

User Action: Determine the reason for the error and correct the command syntax. Refer to Chapter 12 for valid syntax, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

MBXERR, unexpected mailbox error

Facility: NETTRACE

Explanation: An error occurred on the mailbox that trace uses to communicate with the detached trace collector.

User Action: Take corrective action based on the accompanying messages.

MBXREAD, mailbox read timed out

Facility: NETTRACE

Explanation: A read operation timed out on the mailbox used between trace and the detached trace collector process. This situation might occur if the detached trace collector exited abnormally.

User Action: Restart your detached trace process.

Failed to read network mailbox,

Facility: SNATRACE

Explanation: SNATRACE failed to get input from its network mailbox. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

MBXWRITE, mailbox write timed out

Facility: NETTRACE

Explanation: A write operation timed out on the mailbox used between trace and the detached trace collector process. This situation might occur if the detached trace collector exited abnormally.

User Action: Restart your detached trace process.

MISSING, *number* records missing from input files

Facility: NETTRACE

Explanation: The trace sequence being analyzed contains a number of missing records. This situation might occur if a trace file had been deleted.

User Action: None.

NETIO, network communication error

Facility: SNANCP

Explanation: A network error occurred. The secondary system message provides additional information.

User Action: Try again or enter the next command. If necessary, refer to the *OpenVMS System Messages and Recovery Procedures*.

NEWHEADER, new header record encountered

Facility: SNATRACE

Explanation: A new trace file header is read while processing a trace input file. This can happen if you concatenate input files.

User Action: None.

NEW_SEQ, new trace sequence started

Facility: NETTRACE

Explanation: NETTRACE is analyzing a new file that contains a new sequence of trace records. The new sequence is not related to the sequence in the preceding file.

User Action: None.

NMLRSP, listener response - *text*

Facility: SNANCP

Explanation: This is a standard message for text that is passed to SNANCP from the SNANML.

User Action: Take the appropriate action as indicated for each of the following secondary messages:

Bad loopback response

Facility: SNANCP

Explanation: The length or content of a loopback message did not match what SNANCP was expecting.

User Action: Refer to *DECnet SNA Gateway for Synchronous Transport Problem Solving* for information about SNANCP loopback tests. If necessary, have the relevant piece of communications equipment replaced.

Component in wrong state

Facility: SNANCP

Explanation: The component was not in the correct state to receive your request. (For example, a line must be OFF before you can change the parameters for that line.)

User Action: Check the appropriate chapter in this manual to verify the proper command format. Reenter the command.

For DHCF, the message means that the text file cannot be read because the previous READ is not complete. This message might appear after a SET command.

Component in wrong state or wrong parameter

Facility: SNANCP

Explanation: You have specified a parameter that is inappropriate for the command.

User Action: Check the appropriate chapter in this manual for a list of the parameters you can specify and when you can specify these parameters. Reenter the command.

Component not available

Facility: SNANCP

Explanation: The component you specified is either not available or not defined.

User Action: Use the SHOW KNOWN command to verify which components are defined and which state they are in. Reenter the command.

Hardware failure

Facility: SNANCP

Explanation: There is a problem with the hardware.

User Action: If the problem persists, run the diagnostics for the line device.

Incompatible management version

Facility: SNANCP

Explanation: The command is not valid for this version of Gateway software.

User Action: Specify a PU name when you set an LU list parameter for an access name definition.

Invalid identification

Facility: SNANCP

Explanation: The format you specified for the command was incorrect.

User Action: Check the appropriate chapter in this manual to verify the proper command format. Reenter the command.

For DHCF, this message indicates that you specified a PU in the CLEAR command that is not in the DHCF database.

Invalid message format

Facility: SNANCP

Explanation: This is an internal error.

User Action: Report this problem and the circumstances that caused it to your Digital service representative.

Invalid parameter grouping

Facility: SNANCP

Explanation: Some of the parameter changes you requested cannot be specified in the same command as other changes.

User Action: Check the appropriate chapter in this manual for information about the parameters, as well as what state the component should be in when you specify the parameters. Reenter the command, separating the parameters.

Invalid parameter value, *parameter-name*

Facility: SNANCP

Explanation: The value you specified for the parameters was not within the valid range.

User Action: Check the appropriate chapter in this manual to verify the proper command format. Reenter the command.

Line communication error

Facility: SNANCP

Explanation: There was an error in transmitting or receiving loopback data on a line.

User Action: Refer to *DECnet SNA Gateway for Synchronous Transport Problem Solving* to make sure you specified the loopback test correctly. You should also make sure that the modems you are using support loopback tests.

No room for new entry

Facility: SNANCP

Explanation: You have tried to supply too many entries for the parameter.

User Action: Check the appropriate chapter in this manual for the maximum number of entries you can specify for the parameter. Reenter the command.

For channel-attached circuits, this might mean that you have tried to define more than the supported number of circuits for the specified line.

For DHCF, this message means that the SET command has not been performed because the number of defined LUs exceeds the number of MAX LINKS set for the DHCF server (the default MAX LINKS is 16).

Operation failure

Facility: SNANCP

Explanation: A requested operation failed. There is no specific error message associated with the failure. A possible cause for this error is that the Gateway parameters are not defined correctly.

User Action: Check the current Gateway parameters. Refer to Chapter 13 for information about the command you were using. If the problem persists, refer to *DECnet SNA Gateway for Synchronous Transport Problem Solving* or *DECnet SNA Gateway for Channel Transport Problem Solving (OpenVMS and ULTRIX)*.

For DHCF, this means that the SET command has not been performed because it would cause MAX LINKS to become less than the total LUs allocated to DHCF.

Parameter missing

Facility: SNANCP

Explanation: You omitted a required parameter.

User Action: Check the appropriate chapter in this manual for information about the parameters you can specify. Then reenter the command.

Parameter value too long

Facility: SNANCP

Explanation: A parameter that you specified was too long.

User Action: Check the appropriate chapter in this manual for information about the parameter you want to use. Then reenter the command.

Privilege violation

Facility: SNANCP

Explanation: You do not have the necessary privileges to do what you have requested. This message appears when you fail to specify privileged user information with the USE SYSTEM command and then issue a command that requires privileges.

User Action: Ask your system manager to give you the privileged user information that you require.

Resource error

Facility: SNANCP

Explanation: The required resource is not available. This usually occurs when there is a memory shortage.

User Action: It might be necessary to reconfigure the Gateway to make more efficient use of the available resources. Reducing the number of active sessions temporarily gives you more free memory.

Unrecognized component

Facility: SNANCP

Explanation: The component you specified is not known to the node.

User Action: Use the SHOW KNOWN command to check the components that have been defined. Then reenter the command.

Unrecognized function or option

Facility: SNANCP

Explanation: You have requested an action that is not recognized by the local network management function.

User Action: Check Chapter 13 to make sure the command or action that you requested is supported.

Unrecognized parameter type

Facility: SNANCP

Explanation: You have specified a parameter that is not recognized by the network management function.

User Action: Check Chapter 13 for information about the parameters you can use with this command. Then reenter the command.

NOCIRCUIT, circuit is unknown

Facility: SNATRACE

Explanation: The circuit you specified does not exist, or Gateway software is not loaded. There might be a typographical error in your *circuit-id*.

User Action: Enter the correct *circuit-id*. For Gateway-ST, the *circuit-id* is SDLC-*nnn*. For Gateway-CT, the *circuit-id* is CHAN-*nnn*.

NOLINK, logical link terminated

Facility: SNATRACE

Explanation: The logical link between SNATRACE and the server was terminated. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

NOPARLIVE, *parameter* is not valid while tracing live

Facility: NETTRACE

Explanation: You cannot specify *parameter* while live tracing is active.

User Action: None.

NOPARM, no parameters

Facility: SNANCP

Explanation: You did not specify any parameters for the command, or you did not select any parameters in a series of prompts.

User Action: Reenter the command, specifying at least one parameter.

NOPU, physical unit (PU) is unknown

Facility: SNATRACE

Explanation: The physical unit you specified does not exist.

User Action: Use the SNANCP SHOW KNOW PUS command to display a list of known PUs. Reenter the command using one of the known PUs.

NORMAL, normal successful completion

Facility: SNATRACE

Explanation: This message indicates that the action you requested completed successfully.

User Action: None.

NOSESSION, session is unknown

Facility: SNATRACE

Explanation: The session you specified for the trace is not active.

User Action: Use the SNANCP KNOWN LUS command to display a list of LUs that have an active session.

NO_SUCCIR, circuit name not recognized by Gateway node

Facility: SNATRACE

Explanation: You specified a nonexistent circuit.

User Action: Determine a valid circuit name for your operation and try again.

NO_SUCSES, session address not recognized by Gateway node

Facility: SNATRACE

Explanation: You specified a nonexistent session address.

User Action: Determine a valid session address for your operation and try again.

NOTDONE, command not complete

Facility: SNANCP

Explanation: You have not entered enough of the command for SNANCP to be able to complete the requested action.

User Action: Reenter the command using more characters.

NOTINIMSG, received message not INIT

Facility: SNATRACE

Explanation: The server was expecting an initialization message.

User Action: Contact your Digital service representative.

NOTRECORD, message other than "Record Data" received

Facility: SNATRACE

Explanation: System received an unexpected message, not a trace record.

User Action: Contact your Digital service representative.

NOT_SHOW_NODE, cannot show another node while tracing live

Facility: NETTRACE

Explanation: You asked for tracing on one node while while you were tracing on another node. Trace can be connected to only one node at a time.

User Action: None.

NOVALUE, *keyword* keyword requires a value

Facility: SNATRACE

Explanation: You must specify a value for the keyword you entered on the command line.

User Action: Check the command syntax and enter an appropriate value for the command. Refer to Chapter 12, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

OPENERR, error opening *filename*

Facility: NETTRACE

Explanation: An error occurred opening file *filename*.

User Action: Take corrective action based on the accompanying messages.

OPENIN, error opening *file-spec* as input

Facility: SNATRACE

Explanation: You specified a file that could not be opened. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

OPENOUT, error opening *file-spec* as output

Facility: SNATRACE

Explanation: You specified a file that could not be opened. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

PRMLEN, parameter too long *parameter*

Facility: SNANCP

Explanation: You have entered a parameter string or value using too many characters.

User Action: Check the appropriate chapter in this manual to verify the parameter values you can use. Reenter the command.

PRMRNG, parameter value out of range *value*

Facility: SNANCP

Explanation: You used a numeric parameter value that is outside of the valid range.

User Action: Check the appropriate chapter in this manual to verify the valid range of parameter values. Reenter the command.

PRNOTFOUND, process *process-name* not found - use SHOW command for list

Facility: NETTRACE

Explanation: A STOP command failed to find process *process-name*.

User Action: None.

PROCFILE, processing file *filename*

Facility: NETTRACE

Explanation: The file is opened and is being analyzed.

User Action: None.

PROCTRPT, processing *name* tracepoint

Facility: NETTRACE

Explanation: A tracepoint has been found and is now being analyzed.

User Action: None.

PROTERR, protocol error on link to trace collector

Facility: NETTRACE

Explanation: An unexpected protocol error occurred in a message to the trace collector.

User Action: Contact your Digital service representative.

PROTOCOL, protocol error

Facility: SNATRACE

Explanation: The trace record message is smaller than the minimum required length.

User Action: Contact your Digital service representative.

PROT_VERS, unknown protocol version analysis version *n / m*

Facility: NETTRACE

Explanation: The protocol version of the analysis routine that was activated is unknown. It might be incompatible with the version of NETTRACE being used.

User Action: Ensure that your NETTRACE image is compatible with your version of VMS/SNA.

PUNOTAVA, PU has not been activated

Facility: SNATRACE

Explanation: The physical unit to your system was not activated by IBM.

User Action: First make sure that you are using the correct PU. Then ask the VTAM operator to check the line and physical unit (PU) from the IBM host and activate the lines if necessary. If the lines are already activated, there might be hardware problems between your system and the IBM host.

PUNOTSPE, PU name was not specified

Facility: SNATRACE

Explanation: You did not specify the physical unit for the trace.

User Action: See Chapter 12 for information on SNATRACE. Reenter the command.

PUTOOLON, PU name is too long

Facility: SNATRACE

Explanation: The physical unit name you specified contains too many characters.

User Action: Specify a valid PU name and try again.

READCHAR, failed to read device information

Facility: SNATRACE

Explanation: System failed to read device characteristics.

User Action: Take corrective action based on the accompanying messages.

READERR, error reading *file-spec*

Facility: SNATRACE

Explanation: An error occurred while the system was reading the specified file. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

Error reading file *filename*,

Facility: NETTRACE

Explanation: An error occurred reading file *filename*.

User Action: Take corrective action based on the accompanying messages.

RECEIVE, network receive error

Facility: SNATRACE

Explanation: SNATRACE did not receive data from the Gateway. This message might occur with the %SNATRACE-E-REQDATA message. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

Network receive error,

Facility: SNAP

Explanation: Information cannot be received by the remote node because of a problem with the physical or logical link. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages. If necessary, reestablish the problem connection.

REJECT, connect rejected by server

Facility: SNATRACE

Explanation: A request to connect to the SNATRACE server module failed for one of the following reasons:

- The access control information you specified is invalid.
- The partner task exited during the connect sequence.

User Action: Determine the reason for the failure and try to correct the problem. Refer to Chapter 12, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

REPEAT, parameter repeated *parameter*

Facility: SNANCP

Explanation: You repeated a parameter in the command line. A parameter can only appear once in the command line.

User Action: Reenter the command.

REQDATA, request data failed

Facility: SNATRACE

Explanation: SNATRACE did not receive data from the trace server. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

RES_SCROLL, resuming scrolling

Facility: NETTRACE

Explanation: If you are tracing live, scrolling automatically resumes when you finish looking at saved records.

User Action: None.

RETURNED, control returned to process *process-name*

Facility: NETTRACE

Explanation: You logged out of a subprocess and are returned to the parent process.

User Action: None.

RMSERR, error using RMS routines

Facility: NETTRACE

Explanation: An error occurred using RMS routines to access a file.

User Action: Take corrective action based on the accompanying messages.

RTLERROR, unexpected error using run time library routine

Facility: NETTRACE

Explanation: An unexpected error occurred while you were using the run time library routines. Another message follows describing the error.

User Action: Take corrective action based on the accompanying messages.

SEND, network send error

Facility: SNAP

Explanation: Information cannot be sent to the remote node. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

SESIN_USE, trace already in progress for this circuit

Facility: SNATRACE

Explanation: Someone else is running a trace on this circuit.

User Action: Retry using another session address.

SESNOTAVA, session address has not been activated

Facility: SNATRACE

Explanation: The secondary logical unit (SLU) has not been activated from the IBM side.

User Action: Ask the VTAM operator to check the SLU from the IBM host and activate it if necessary.

SETUP, failed to enable Ctrl/C

Facility: SNATRACE

Explanation: SNATRACE could not enable Ctrl/C interception.

User Action: Take corrective action based on accompanying messages.

SHOFIL, unable to open output for show

Facility: SNANCP

Explanation: RMS returned an error when attempting to open the output file for the SHOW commands. The RMS error is displayed for additional information.

User Action: For information on the RMS error message, refer to the *OpenVMS System Messages and Recovery Procedures*.

SHOIO, error writing output file for show

Facility: SNANCP

Explanation: RMS returned an error when attempting to write to the output file for the SHOW commands. The RMS error is displayed for additional information.

User Action: For information on the RMS error message, refer to the *OpenVMS System Messages and Recovery Procedures*.

SHOW_DONE, SHOW command complete - press Return to continue

Facility: NETTRACE

Explanation: A SHOW command has completed.

User Action: Press RETURN to remove the current show display and restore the previous display.

SHOW_OPTION, specify SHOW TRACE_COLLECTOR or SHOW KEY

Facility: NETTRACE

Explanation: To use the SHOW command, specify TRACE_COLLECTOR or KEY.

User Action: See Chapter 11 for details.

SHOW_PEND, SHOW command is already pending

Facility: NETTRACE

Explanation: You issued a SHOW command with a previous SHOW command still outstanding.

User Action: Wait for the first command to complete before issuing the next SHOW command.

SMGERR, unexpected error using screen management routines

Facility: NETTRACE

Explanation: An unexpected error occurred while you were using screen management routines.

User Action: Take corrective action based on the accompanying messages.

SNACOMM, SNA communication error

Facility: SNATRACE

Explanation: The trace server encountered an error when retrieving trace data. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

SNDINIT, failed to transmit initiate message

Facility: SNATRACE

Explanation: SNATRACE failed to start a trace operation. This message is usually accompanied by additional messages indicating the reason for failure and generally indicates a logical link failure.

User Action: Take corrective action based on the accompanying messages.

SRVLNKABO, link aborted by server (*octal-number*)

Facility: SNATRACE

Explanation: The logical link between the SNATRACE utility and the SNATRACE server was terminated by the server because of an SNATRACE software error.

User Action: Record the information displayed and report the error to your system manager. Report the problem to your Digital service representative.

START, failed to receive start message

Facility: SNATRACE

Explanation: SNATRACE failed to start a trace operation. This message is usually accompanied by additional messages and generally indicates a logical link failure.

User Action: Take corrective action based on the accompanying messages.

STARTED, tracing has been started

Facility: NETTRACE

Explanation: Tracing has been started. The tracepoint specified must match the actual lines being traced exactly for trace data to be collected.

User Action: None.

STARTPEND, previous START command pending

Facility: NETTRACE

Explanation: A previous start command has not completed.

User Action: Contact your Digital service representative.

STOP_RECORD, TRACE_CAPTURE has stopped recording

Facility: NETTRACE

Explanation: The trace capture module has stopped recording, either because of an error or because the communications software being traced has been turned off.

User Action: None.

SYNTAX, command syntax error *keyword*

Facility: SNANCP

Explanation: You have not entered the command in the correct format.

User Action: Check the appropriate chapter in this manual to verify the proper command format. Reenter the command.

Error parsing *string*,

Facility: SNATRACE

Explanation: A syntax error was detected. This type of error is generally caused by a typographical error.

User Action: Refer to Chapter 12 for the correct syntax, or enter HELP at the SNATRACE> prompt for information.

SYSIO, unexpected I/O error

Facility: NETTRACE

Explanation: An unexpected I/O error has occurred.

User Action: Take corrective action based on the accompanying messages.

TINY, the screen needs to be at least 24 x 80

Facility: SNAP

Explanation: The screen is too small to display the SNAP screen. The screen needs to be at least 24 rows by 80 columns.

User Action: Use the DCL SET TERMINAL command to define a larger screen area.

TRALOAFAI, fail to load translation table from file *file-spec*

Facility: SNATRACE

Explanation: Read the secondary message.

User Action: Take the action detailed in the secondary message.

TRANSMIT, network transmit error

Facility: SNATRACE

Explanation: SNATRACE did not send a message to the trace server. Typically, this error occurs during trace initialization. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages. Reenter the command.

TRPT_NOT_FOUND, tracepoint *tracepoint-name* not found

Facility: NETTRACE

Explanation: Tracepoint *tracepoint-name* is not a valid tracepoint name.

User Action: Specify a valid tracepoint name (see Chapter 11).

TRUNC, data truncated

Facility: SNATRACE

Explanation: The trace data was truncated because the SNATRACE server did not have enough buffer space to accommodate the entire message.

User Action: Increase the size of available buffers with the /SIZE qualifier when you redo the trace operation.

UNKNOWN_TRPT, unknown tracepoint protocol type found

Facility: NETTRACE

Explanation: A trace record with an unknown protocol type was found.

User Action: Contact your Digital service representative.

UNLIKELY, unexpected system service failure

Facility: SNAP

Explanation: System service failed for an unknown reason. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

UNRCMD, unrecognized command *command*

Facility: SNATRACE

Explanation: SNATRACE does not recognize or support the command you entered. The command was probably specified incorrectly.

User Action: Refer to Chapter 12 for the correct command, or enter HELP at the SNATRACE> prompt for information. Reenter the command.

UNSVRS, version of listener on executor is not supported

Facility: SNANCP

Explanation: The Gateway Network Management Listener (SNANML) did not respond with a valid version number. The executor cannot be accessed by network management.

User Action: None. You cannot connect to the node.

USING, using system *node-name*

Facility: SNANCP

Explanation: This message is displayed when SNANCP uses the information in logical name SNAGM$GATEWAY to determine which Gateway to connect to.

User Action: None. This is an informational message.

VALSET, value of *parameter* set to *value*

Facility: NETTRACE

Explanation: A parameter had its value changed to the value shown. The value that you requested lies outside the permitted range, or the value chosen for another parameter affects this parameter.

User Action: None.

VERSKEW, version skew detected connecting to trace collector

Facility: NETTRACE

Explanation: NETTRACE and the trace collector software are operating incompatible protocol versions.

User Action: Refer to the product SPDs, and check that the software on the two systems is compatible.

WRITEERR, error writing *file-spec*

Facility: SNATRACE

Explanation: SNATRACE could not write to the file specified. This message is usually accompanied by additional messages indicating the reason for failure.

User Action: Take corrective action based on the accompanying messages.

Error writing file *filename*,

Facility: NETTRACE

Explanation: An error occurred writing a record to *filename*.

User Action: Take corrective action based on the accompanying messages.

D

Event Messages

A DECnet host node can record event-logging information generated by the Gateway. You can use NCP and SNANCP commands to configure the event logger so that you can do the following:

- Use a monitor process to send event messages to OPCOM, which displays event messages on operator terminals and records them in the operator log.
- Use a logging console to write events in text form to a file or a dedicated terminal.
- Store the information in a file in binary form. This information is for the use of Digital software services specialists.

The information you receive from the event logger is in the form of event messages. Events are defined according to codes, which identify the layer or resource to which the event applies. Event messages have severity levels indicating the importance of the event. Chapter 10 explains how you can control what level of event messages are logged and where the messages are logged.

The usual method of logging is to use a logging monitor to send events to OPCOM, which will then display the event messages on operator terminals. You may also want to record events in a text file by specifying a logging console in your logging commands. This will, for example, allow you to separate Gateway event messages from all of the other DECnet event messages that go to OPCOM. Use the third possibility, that of logging events to a file in binary format, only if you are giving the file to a Digital Software Services Specialist.

To receive event messages at an operator terminal, you need to do the following:

1. Specify MONITOR as the logging component for event messages.
2. Specify the sink node where the messages are logged.
3. Specify which levels of messages are logged.

4. Set event logging ON.
5. Make sure you have OPER privilege.
6. Set the required terminal as a system terminal.

The first four steps are explained in Chapter 10. The default is for messages to be logged at the Gateway load host. The *Guide to Maintaining an OpenVMS System* tells you about system privileges so that you can ensure that you have OPER privilege.

A system program called OPCOM logs events. When you have decided at which terminal you want OPCOM to display the messages (the terminal can be a console), enter the following command from that terminal in order to set it as a system terminal:

```
$ REPLY/ENABLE=NETWORK
```

You then receive all network event messages at that terminal. All Gateway event messages are network messages. You receive other DECnet network messages as well as the Gateway messages. Gateway event messages are identified by the text "DECnet SNA".

When using the Gateway, event messages logged by OPCOM have the following format:

DECnet SNA event *nnn.nn event-text*

From node *node-address (node-name), dd-mmm-yyyy hh:mm:ss.cc*

component component-id, Severity = *level*

message1_text

message2_text

where:

nnn.nn	Is the DECnet or Gateway event code. All Gateway event messages have a code greater than 255.
event-text	Is the text that always appears with the event class and type.
node-address (node-name)	Is the Gateway node address and node name.
dd-mmm-yyyy	Is the date on which the event occurred.
hh:mm:ss.cc	Is the time at which the event occurred.

component component-id	Is the name and ID of the Gateway component sending the event messages.
level	Is the severity level of the event.
message1_text and *message2_text*	Are messages giving you the details of the event. The second of these messages can be an OpenVMS system message. For information on OpenVMS system messages refer to the *OpenVMS System Messages and Recovery Procedures*. The text can contain SNANCP error messages. See Appendix C.

The messages in this appendix are listed according to their event code.

D.1 Line Events—256.*x*

DECnet SNA event 256.0, Line state change

From node *address (name), date-time*

Line *ID*, Severity = information or warning

A message follows explaining the change.

DECnet SNA event 256.1, Channel reset

From node *address (name), date-time*

Line *ID*, Severity = error

Type = system or selective

DECnet SNA event 256.2, Error on channel

From node *address (name), date-time*

Line *ID*, Severity = error

One of the following messages is shown as part of the event message:

- Unsupported command byte
- Bus-out parity error

DECnet SNA event 256.3, Device or driver error

From node *address (name), date-time*

Line *ID*, Severity = fatal

The following message is shown as part of the event message:

- Device timeout

D.2 Circuit Events—257.*x*

DECnet SNA event 257.0, Circuit state change

From node *address (name), date-time*

Circuit *ID*, Severity = information or warning

A message follows explaining the change.

DECnet SNA event 257.1, Packet format error

From node *address (name), date-time*

Circuit *ID*, Severity = error

Packet header = *packet*

If an oversize packet arrives on the Gateway, the link header and the first 6 bytes (usually the transmission header) of the packet are logged.

D.3 PU Events—258.*x*

DECnet SNA event 258.0, PU state change

From node *address (name), date-time*

PU *ID*, Severity = information

A message follows explaining the change.

D.4 Session Events—259.*x*

DECnet SNA event 259.0, LU state change

From node *address (name), date-time*

LU *ID*, Severity = information

A message follows explaining the change.

DECnet SNA event 259.1, SNA session protocol error

From node *address (name), date-time*

LU *ID*, Severity = warning

A message follows explaining the protocol error.

DECnet SNA event 259.2, Access denied

From node *address (name), date-time*

LU *ID*, Severity = warning

Node = *name*, user = *name*, terminal = *ID*

This message indicates that an attempt was made to access the LU by an unauthorized user. The node, user, and terminal parameters identify the process which was trying to use the restricted LU. (The terminal name is not shown if not applicable. Also, not all access routines currently supply a user name and a terminal name.)

D.5 Counter Events—265.*x*

DECnet SNA event 265.0, Automatic counters

From node *address (name), date-time*

component ID, Severity = information

The *component* for this event message can be a line, circuit, or PU. The message is followed by the counters for that component.

DECnet SNA event 265.1, Counters zeroed

From node *address (name), date-time*

component ID, Severity = information

The *component* for this event message can be line, circuit, PU, LU, or server. The message informs you that the component counters have been set to zero. The message is followed by the counters for that component.

DECnet SNA event 265.2, Counter overflow

From node *address (name), date-time*

component ID, Severity = information

The *component* for this event message can be line, circuit, PU, LU, or server. The message informs you that one of the component's counters has overflowed. The name of the counter in question is shown.

D.6 Initialization Events—266.*x*

DECnet SNA event 266.0, Gateway initialization

From node *address (name), date-time*

SNA Gateway management, Severity = information

This message gives you information about the status of Gateway initialization and includes one of the following information messages:

- Started, *Gateway ID*
- Complete, status = success
- Complete, status = failed

DECnet SNA event 266.1, Initialization failure

From node *address (name), date-time*

SNA Gateway management, Severity = error

This message tells you that the Gateway initialization failed and includes one of the following error messages:

- Failed to connect, load host NML
- Failed to connect, local NML
- Failed to connect, load host SNAINI
- Error loading node database
- Version mismatch, load host NML
- Version mismatch, local NML
- Retry limit exceeded
- Local resource error
- Link broken, local NML
- Link broken, load host NML
- Link broken, load host SNAINI

These messages might be followed by an OpenVMS system message.

DECnet SNA event 266.2, Initialization message

From node *address (name), date-time*

SNA Gateway management, Severity = information

This event is used to display a message issued by the SNAINI process on the load host. One of the following messages might appear:

- Default node database loaded
- Setting Gateway DECnet configuration
- Setting Gateway SNA configuration
- An error message from SNAINI

D.7 Logging Events—266.*xx*

DECnet SNA event 266.10, Logging sink state change

From node *address (name), date-time*

Logging sink type = file or monitor, Severity = information or warning

This message indicates that there is a change in the event logging state. Subsequent messages give the sink node number, the sink node name, and the logging state change.

DECnet SNA event 266.11, Event logging sink failure

From node *address (name), date-time*

Logging sink type = file or monitor, Severity = warning or error

This message indicates that there is a failure in the event logging. Subsequent messages might give the sink node number, the sink node name, and/or an OpenVMS system message.

D.8 SNA-ACCESS Server Events—268.0

DECnet SNA event 268.0, Gateway Access server message

From node *address (name), date-time*

Server SNA-ACCESS, Severity = *level*

The severity *level* can be information, warning, error, or fatal. Up to two of the following messages are shown as part of the event message:

- Information:
  - Server state change, *secondary message*
  - Node *node*, user *user*, terminal *terminal* connected to LU *lu-id*

- Node *node*, user *user*, terminal *terminal* disconnected from LU *lu-id*

- Warning or Information:
  - Maximum allowable links (*n*) in use
  - Link aborted, node = *node*, user = *user*, terminal = *terminal*, LU = *lu-id*, *secondary message*

    The terminal name and user name are not shown if not applicable.

    The following are possible secondary messages:

Note

While the following secondary messages do exist, the current software may not be capable of displaying every message listed here.

- Link aborted by user
- BIND rejected by user, sense data = *sense data*
- Incompatible protocol versions
- Insufficient Gateway resources
- No such access name
- PU name not specified
- No such PU
- PU not activated
- PLU name not specified
- No such session address
- Session address not activated
- Session address already in use
- Connection rejected by SSCP or PLU, sense data = *sense data*
- BIND values unacceptable to Gateway
- UNBIND received, session unbound
- DACTLU received, session deactivated
- GAP violation detected by Gateway
- Internal error in Gateway

- Link or PU reinitialized
- Abnormal session termination
- No session address available
- Access to LU denied
- GAP violation detected by AI
- Internal error in AI

- Error:
  - IBM exceed maximum receive buffer size of *nn* bytes

    An IBM application sent a message that is larger than the RU size specified for the Gateway. You must use a larger RU size.
  - Unknown GAP message received, code = *code*, PLU = *name*, SLU = *SNA-n.nnn*, node = *node*, user = *username*

    Contact your Digital Support Center.
  - Illegal server event *nn* while in state *nn*, node = *nodename*, user = *username*, terminal = *terminal-id*, LU = *LU-id*

    Contact your Digital Support Center.
- Fatal:
  - Declaration channel deleted
  - Server failed to declare network object

D.9 DHCF Server Events—268.10

DECnet SNA event 268.10, DHCF Server message

From node *address (name), date-time*

Server DHCF, Severity = *level*

The severity *level* can be information, warning, or error. The messages that can be displayed as part of this event are listed below:

- Information:
  - Logon text loaded, file = *filespec*
  - Server state change, *secondary message*
  - LU *LU-id*, connected to OpenVMS node *node-name*
  - LU *LU-id*, requesting connection to OpenVMS node *node-name*

- LU *LU-id*, session deactivated by SSCP
- LU *LU-id*, session aborted
- LU *LU-id*, session unbound

• Warning—The failure does not cause any existing sessions to terminate:

- Failed to load login text file, file = *filespec*

  There is an additional line describing the message that was returned by RMS.
- LU *LU-id*, DECnet connect failed, node = *node*

  There is an additional line describing the message returned by DECnet.
- LU *LU-id*, DECnet connect rejected, unknown reason
- LU *LU-id*, DECnet link terminated, node = *node*

  There is an additional line describing the message returned by DECnet.
- LU *LU-id*, DECnet connect rejected, incompatible versions
- LU *LU-id*, DECnet connect rejected, invalid LU type
- LU *LU-id*, DECnet connect rejected, invalid screen size
- LU *LU-id*, DECnet connect rejected, no channels available
- LU *LU-id*, (PSA) Negative rsp to normal request, sense code: *IBM sense-code*
- LU *LU-id*, "listen" operation failed

  There is an additional line describing the message returned by LU services.
- LU *LU-id*, unacceptable field in BIND

• Error messages—The failure causes the termination of a session, or it indicates a serious problem that needs correcting. Refer to the DHCF documentation.

- LU *LU-id*, bracket protocol violation when receiving
- LU *LU-id*, DFC chaining error when receiving
- LU *LU-id*, DFC chaining error when sending
- LU *LU-id*, DFC error when receiving expedited rsp
- LU *LU-id*, DFC error when receiving normal request

- LU *LU-id*, DFC error when receiving normal rsp
- LU *LU-id*, DFC error when sending normal request
- LU *LU-id*, DFC half-duplex error when sending
- LU *LU-id*, DFC half-duplex error when receiving
- LU *LU-id*, DFC immediate rsp violation when receiving
- LU *LU-id*, DFC immediate rsp violation when sending
- LU *LU-id*, expedited -RSP received, sense code = *IBM sense-code*
- LU *LU-id*, FMD status error
- LU *LU-id*, HCF rejected all BIND images
- LU *LU-id*, invalid category for received expedited RU
- LU *LU-id*, invalid category for received RU
- LU *LU-id*, invalid expedited request received
- LU *LU-id*, invalid host-DHCF protocol byte: *byte*
- LU *LU-id*, negative rsp to expedited request, sense code = *IBM sense-code*
- LU *LU-id*, negative rsp to normal request, sense code = *IBM sense-code*
- LU *LU-id*, normal -RSP received, sense code = *IBM sense-code*
- LU *LU-id*, S/W error, input *hex* in state *hex*
- LU *LU-id*, TC error in received sequence number
- LU *LU-id*, TC immediate rsp violation when sending or receiving
- LU *LU-id*, unexpected DFC expedited request
- LU *LU-id*, unexpected DFC normal request
- LU *LU-id*, unexpected expedited +RSP received
- LU *LU-id*, unexpected HCF-DHCF message
- LU *LU-id*, unexpected normal +RSP received
- LU *LU-id*, wrong device type, not LU type 2

D.10 RJE Server Events—269.0

DECnet SNA event 269.0, RJE Server message

From node *address (name)*, *date-time*

Server RJE, Severity = *level*

The severity level can be information, warning, or fatal. Up to two of the following messages are shown as part of the event message:

- Information:
  - Server state change, *secondary message*
  - Connection request from node *node*
- Warning:
  - Maximum allowable workstations (*n*) in use
  - Maximum allowable links (*n*) in use
- Fatal:
  - Server failed to declare network object
  - Declaration channel deleted

Index

D

E

M

N

O

P

R

S

T

U

Z