NS 3000/iX Operations and Maintenance Reference Manual

HP e3000 MPE/iX Computer Systems

Edition 8

Manufacturing Part Number: 36922-90042
E0801

U.S.A. August, 2001

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

Acknowledgments

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Contents

Contents

Contents

Contents

Figures

Figures

Tables

Tables

Preface

Network Services for MPE/iX based systems are provided by an HP data communications product named NS 3000/iX. This manual describes the system-level commands and utilities used to perform network operations, maintenance, and troubleshooting after the initial network configuration.

NS 3000/iX enables your HP e3000 to communicate with other HP computer systems as part of a distributed network. These systems can be other HP e3000s, HP 9000s, HP 1000s, and PCs. Networks operating over NS 3000/iX links can be interconnected. When two or more networks are connected in this manner, the resulting network is called an internetwork. A network or internetwork can be created using one of the following NS 3000/iX links:

- NS Point-to-Point 3000/iX Link
- X.25 iX System Access
- ThinLAN 3000/iX Link (includes the ThickLAN option for coaxial cable) and EtherTwist option for twisted-pair wiring.
- Token Ring/iX Link
- Fiber Distributed Data Interface/iX Link
- HP-PB 100VG-AnyLAN Network Adapter
- HP-PB 100Base-T Network Adapter

Special Note

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP e3000 line of computers.

In HP documentation and in talking with HP e3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Intended Audience of this Manual

This manual is intended for those with data communications experience. Also required is knowledge of the MPE/iX operating system at the system supervisor level, and a familiarity with the SYSGEN dialogue, resource management, and console commands.

Organization of the Manual

You can use this manual as either a command reference or a user's guide, depending on your needs. It contains the following sections:

- Chapter 1 , “Overview of NS 3000/iX Operations,” provides a general

description of the operations and maintenance functions required for your NS network. It includes a table that can help you determine which tool to use to obtain specific information about your network.

- Chapter 2 , “Operating Your Network,” provides step-by-step instructions for starting and stopping network links and services and for verifying network connections and services.
- Chapter 3 , “Getting Information About the Network,” provides step-by-step instructions for obtaining information about software version numbers, network configuration, link status, network status, Network Services status, and network performance.
- Chapter 4 , “Troubleshooting Process,” provides a generalized problem-solving strategy for identifying problems with the network’s operation.
- Chapter 5 , “Common Network Problems,” provides strategies for dealing with interactive or programmatic problems, command errors, nodal problems, link problems, and software problems.
- Chapter 6 , “Using NETTOOL,” describes the NETTOOL diagnostic utility and provides step-by-step instructions for using each of the available diagnostics within NETTOOL.
- Chapter 7 , “Commands,” describes the MPE/iX commands for NS 3000/iX link products.
- Appendix A , “LINKCONTROL Command,” defines the fields output by the LINKCONTROL STATUS command and its associated parameters.
- Appendix B , “Submitting an CR,” describes how to submit a Change Request (CR) and forward it to your HP Service Representative.

Related Publications

The following manuals may be of interest to you when working with the NS 3000/iX network services and link products:

For the NS 3000/iX Links

- *HP e3000/iX Network Planning and Configuration Guide*
- *Configuring and Managing Host-Based X.25 Links*
- *Using the Node Management Services (NMS) Utilities*
- *NS 3000/iX NMMGR Screens Reference Manual*
- *HP SNMP/XL User’s Guide*
- *Berkeley Sockets/iX Reference Manual*
- *NetIPC 3000/XL Programmer’s Reference Manual*
- *HP36923A LAN 3000/iX Link and Terminal LAN Link Hardware Reference Manual*

- *LAN Cable and Accessories Installation Manual*
- *Central Bus Programmable Serial Interface Installation and Reference Guide*
- *HP 28663A EtherTwist Hub Installation Guide*

For the NS 3000/iX Services

- *Using NS 3000/iX Network Services*

For Either the NS 3000/iX Links or Services

- *NS 3000/iX Error Messages Reference Manual*

For the Distributed Terminal Subsystem (DTS)

- *Configuring Systems for Terminals, Printers, and Other Serial Devices*
- *Using the OpenView DTC Manager*

1 Overview of NS 3000/iX Operations

In the daily operations of the NS 3000/iX network communications products, you will need to perform a number of management tasks. These tasks include starting and stopping Network Services or links, verifying network connections, obtaining status information, and troubleshooting problems that might occur on the network.

Hewlett-Packard provides a number of tools to help you in performing these functions. The tools are included with the networking software and their use is detailed in this manual.

This chapter provides an overview of the normal operations and maintenance functions required for your NS network. It includes general information on the following topics:

- Creating your network.
- Operating your network.
- Obtaining information about your network.
- Troubleshooting your network.

Creating Your Network

This manual assumes that you have a functional network with at least one NS 3000/iX link properly configured to allow data communications to occur between systems. If you have not yet created your network, or if you need to make modifications to your network configuration, you will need to use the *HP e3000/iX Network Planning and Configuration Guide*. An overview of the procedures required to plan and configure the network is provided below.

Network Creation Overview

1. Check that the hardware components required for NS 3000/iX have been installed and verified according to the procedures in the hardware installation manuals listed in the preface of this manual.
2. Use the software verification procedures described in this manual to check that the data communications software has been installed properly.
3. Plan your network configuration by filling out the worksheets provided in the *HP e3000/iX Network Planning and Configuration Guide*.
4. Configure the transport and links by using the `NMMGR.PUB.SYS` utility to update the configuration file and, if required, the network directory file.
5. Validate the network transport to check for consistency of configuration values.
6. Cross-validate the `NMMGR` configuration file with the system configuration files within `SYSGEN`.

The NS 3000/iX Products

An NS 3000/iX network consists of one or more of the available NS 3000/iX link products configured to allow communications between systems (nodes) on the network. The Network Services are available to allow users to perform applications across the network.

Your network will include one or more of the following link products:

- **ThinLAN 3000/iX Link**. Supports IEEE802.3/Ethernet LAN connections.
- **Token Ring/iX**. Supports IEEE802.5 token ring LAN connections.
- **NS Point-to-Point Network Link/iX**. Supports LAP-B connections over leased lines or switched auto-dial lines.
- **X.25 iX System Access**. Supports connections to X.25 public or private data networks.
- **Fiber Distributed Data Interface/iX**. Provides a single-attach connection to an FDDI network through an FDDI concentrator.
- **HP-PB 100VG-AnyLAN Network Adapter.** Connects an HP e3000 computer using the HP-PB backplane to a 100VG-AnyLAN network.
- **HP-PB 100Base-T Network Adapter.** Connects an HP e3000 computer using the HP-PB backplane to a 100Base-T network.

The following Network Services are available to run over the NS 3000/iX link products:

- **Virtual Terminal (VT)**. Creates an interactive session on another system in the network. Multiple concurrent sessions are possible.
- **Remote File Access (RFA)**. Allows I/O operations to files and devices on remote systems.
- **Remote DataBase Access (RDBA).** Allows access to turboIMAGE databases on remote nodes.
- **Network File Transfer (NFT)**. Allows transfer of files from one node to another.
- **Remote Process Management (RPM)**. Allows the creation and termination of processes on remote nodes.

Operating Your Network

You perform most of the daily operations required by the NS network through the use of the provided NS 3000/iX network commands. Complete information on the syntax and function of all the commands is contained in Chapter 7 , “Commands,” of this manual.

The NETCONTROL START command allows you to initiate the network transport as well as the individual networks on an active transport. The NETCONTROL STOP command allows you to stop an individual NI or all active entities of the network transport.

To start and stop the Network Services after the network is started, you use the NSCONTROL START and NSCONTROL STOP commands. The NSCONTROL ABORT command is useful when you need to terminate services immediately.

See Chapter 2 , “Operating Your Network,” for step-by-step instructions for using these commands.

Obtaining Information About Network

You may need to obtain many different kinds of information about your network and its operations. This information ranges from the version numbers of the software modules you are running to complete statistical summaries of events that are taking place on a specific link. The tools and commands that are available to help you obtain the various types of information are summarized in Table 1-1. Once you have determined which command or tool you need to use, you can refer to the detailed information about that tool in a later chapter of this manual.

Table 1-1 Tools for Obtaining Network Information

To Get Information About...	Use...	Refer to...
Active log file	SHOWNMLOG	Chapter 7
Active Network Services	NSCONTROL STATUS	Chapter 3, 7
Active NIs	NETCONTROL STATUS STATUS NETTOOL	Chapter 3, 7 Chapter 3, 6
Buffers	NETCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 3, 6
Configuration	CONFIGURATION SUMMARY NETTOOL	Chapter 3, 6
Configuration (link)	LINKCONTROL	Chapter 3, 7
Configuration (network interface)	NETCONTROL STATUS	Chapter 3, 7
Control requests	LINKCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Devices	NETCONTROL STATUS STATUS NETTOOL	Chapter 7 Chapter 6
Exception requests	LINKCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Header data	NETCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Home network	NETCONTROL STATUS	Chapter 3, 7
Linkname	LINKCONTROL STATUS NETTOOL	Chapter 3, 7 Chapter 6
Link statistics	LINKCONTROL STATUS NETTOOL	Chapter 3, 7 Chapter 6
Link type	LINKCONTROL STATUS NETTOOL	Chapter 3, 7 Chapter 6

Table 1-1 Tools for Obtaining Network Information

To Get Information About...	Use...	Refer to...
Log file	SHOWNMLOG	Chapter 7
Messages	NETCONTROL TRACE NMDUMP NETTOOL	Chapter 6
Network interface type	NETCONTROL STATUS STATUS NETTOOL	Chapter 3, 7 Chapter 6
Network directory	CONFIGURATION SUMMARY NETTOOL	Chapter 3, 6
Network Services	NSCONTROL STATUS	Chapter 3, 7
Network Services events	NSCONTROL LOG NMDUMP NETTOOL	Chapter 7 Chapter 6
Network Services servers	NSCONTROL STATUS	Chapter 3, 7
Network Services users	NSCONTROL STATUS	Chapter 3, 7
NI protocols	NETCONTROL STATUS	Chapter 3, 7
NI start and stop times	NETCONTROL STATUS	Chapter 3, 7
Nodal management events	NETCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Node name	NETCONTROL STATUS STATUS NETTOOL	Chapter 3, 7 Chapter 6
Number of bytes transmitted	LINKCONTROL STATUS NETTOOL	Chapter 7 Chapter 6
Number of frames transmitted	LINKCONTROL	Chapter 7
Performance	X25CHECK PING LOOPINIT RESOURCE MONITOR XPPERF	Chapter 3, 6 Chapter 3, 6 Chapter 3, 6 Chapter 6 Chapter 6
Protocol Statistics	STATUS NETTOOL	Chapter 6
Read and write requests	LINKCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Round trip response times	LOOPINIT NETTOOL	Chapter 3, 7
State transitions	NETCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6
Status requests	LINKCONTROL TRACE NMDUMP NETTOOL	Chapter 3, 7 Chapter 6

Table 1-1 Tools for Obtaining Network Information

To Get Information About...	Use...	Refer to...
Transmission errors	LINKCONTROL STATUS NETTOOL	Chapter 3, 7 Chapter 6
Version numbers (network transport)	NETCONTROL VERSION	Chapter 3, 7
Version numbers (network services)	NSCONTROL VERSION	Chapter 3, 7
Version numbers (NMS)	NMMAINT.PUB.SYS	Chapter 3
X.25 connection response	X25CHECK NETTOOL	Chapter 3, 6
X.25 connection statistics	X25STAT NETTOOL	Chapter 3, 6

Troubleshooting Network

From time to time you may experience problems on your network. Often, you can easily diagnose and correct the problems without the need to contact a Hewlett-Packard service representative. You may choose to perform some troubleshooting operations on your own, following the guidelines provided in this manual.

Troubleshooting Process Overview

When problems occur, it is helpful to follow a logical process of characterizing and identifying the problem so that you can determine the appropriate course of action to take. Chapter 4 , “Troubleshooting Process,” of this manual suggests a process to follow when troubleshooting problems with network connections.

Chapter 5 , “Common Network Problems,” summarizes most of the common types of problems that might occur on your network. Once you have identified the problem following the process described in Chapter 4 , “Troubleshooting Process,” use Chapter 5 , “Common Network Problems,” as a guide to the actions you can take to resolve the problem.

If you are unable to identify or resolve a network problem using the information provided in Chapter 4 , “Troubleshooting Process,” and Chapter 5 , “Common Network Problems,” of this manual, consult your Hewlett-Packard service representative.

Tools

Hewlett-Packard provides a range of tools to help you perform line verification or run software or hardware diagnostics. You run most of the network diagnostic tools using the NETTOOL utility. NETTOOL provides a set of core functions and an interface for running software and line verification tools that were run standalone on earlier versions of MPE. Help information is available when you use NETTOOL. You can also develop your own tools to run as part of the NETTOOL package.

See Chapter 6 , “Using NETTOOL,” for a complete discussion of NETTOOL and instructions for using each of the NETTOOL diagnostics.

2 Operating Your Network

During normal operations of your network, you will need to do little more than to start and stop the network links and services. You may also need to verify a communications link or perform a quick validation of a Network Service.

This chapter provides instructions for using the provided tools or commands to perform the following operations:

- Start network links and services.
- Verify network connections and services.
- Stop network links and services.

Starting Links and Services

You use the NETCONTROL START command to start the network links and the NSCONTROL START command to start the Network Services. You must start at least one link before you can start Network Services.

When you start the first link, the network transport is initiated as well. The **network transport** is the software responsible for sending data out over the appropriate communications links, receiving incoming data, and routing incoming or outgoing data to the appropriate destinations. It includes the general protocols and the network interfaces.

Start a Network Link

You will need to issue a start command for each network interface (NI) you want to activate. The NET and GATE keywords allow you to specify the NI that you want to start. Use NET with the network interface name associated with the link you want to initiate. Use GATE with the network interface name of the gateway half you want to initiate.

The first NETCONTROL START command you enter will initiate the network transport software, including the configured protocols. You can initiate the network transport by itself by entering the NETCONTROL START command without any keywords, but you will not be able to communicate with any other nodes on the network.

The NETCONTROL START command requires NM capability.

Start a Link

Issue the following command to start an NS link:

```
NETCONTROL START;NET=NIname
```

The *NIname* is the network interface name that you configured through NMMGR. Start other links as required by entering the command using the appropriate NI names.

Start a Host-Based X.25 Link

If your network includes X.25 links and you are using host-based network management, you will need to use the DTCCNTRL command before you issue the NETCONTROL START command for the X.25 NI. DTCCNTRL starts X.25 and PAD support for the DTC/X.25 Network Access card. Issue the following command (System Operator capability required):

```
DTCCNTRL DTC=dtcname;CARD=cardnumber;FUNC=function
```

where *function* is one of the following:

STARTX25 to start X.25 services;

STARTPADSUP to start PAD support services;

STARTBOTH to start X.25 and PAD support services.

For more information on starting host-based X.25 links as well as other uses of the DTCCNTRL command, see *Configuring and Managing Host-Based X.25 Links*.

NOTE If you are starting an X.25 link for a system using PC-based network management or if you are not starting an X.25 link, you do not need to use the DTCCNTRL command.

Start Network Loopback

You must start the loopback NI if you wish to perform local loopback or to DSLINE to the local node. Enter the following command:

```
NETCONTROL START;NET=loopbackNIname
```

where *loopbackNIname* is the name configured for the loopback NI. LOOP is the default name.

NOTE If you use guided configuration to create any NI, a loopback network interface, whose NI name is LOOP, is automatically generated.

Start Network Services

You can start all the available Network Services with a single command, or you can start one or more individual services. To start all Network Services enter the following command:

```
NSCONTROL START
```

To start one or more of the individual services, enter the command followed by an equal sign and a list of the desired services, separated by commas. For example, to start only Virtual Terminal and Reverse Virtual Terminal for users on remote nodes, enter the following command:

```
NSCONTROL START=VT,VTR
```

See Chapter 7 , “Commands,” for complete details on the use and syntax of NETCONTROL START and NSCONTROL START.

Verifying Network Connections and Services

Several line verification tests are available to help you verify the operation of NS 3000/iX services and link products.

NSLOGON establishes temporary connections to other nodes to verify that the network transport is operating correctly between the two nodes using the connection.

XPVAL is an interactive test that uses the NetIPC intrinsics to make sure that the network transport is working correctly.

QVALNS and **NSTEST** both perform a quick validation of the Network Services. QVALNS runs through a job while NSTEST runs interactively.

You can run all of these tests either standalone or through the NETTOOL utility. Hewlett-Packard suggests that you run them through the NETTOOL utility to take advantage of its facilities, including online help.

Check the Network Transport

Perform the following steps to use XPVAL to check the network transport. (Note that you may also use NSLOGON to establish a temporary connection between nodes to check the network services. See Chapter 6 , “Using NETTOOL,” for more information.)

1. Make sure the network transport is active on this node and on any other node that will be a part of this test.
2. Run the NETTOOL utility by entering the program name:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.
3. Enter XPVAL to run the transport validation.
4. XPVAL will run a local program (XPVALLOC) and will prompt you for the information it needs to perform the validation. To check the local transport, enter information about the local loopback NI.
5. To check the transport between the local node and a remote node, make sure XPVAL is running on the remote node as well and enter information about the remote node.
6. XPVAL will run a one minute connection test to verify the operation of the transport and report any errors it encounters. See information on the error messages.

XPVAL Line Test Error Messages

Error messages for the XPVAL line tests appear in inverse video at the system console. Some errors allow the test to continue, so they may scroll off the top of the terminal screen. Copy the error message information for further diagnosis.

Error Message Categories

Errors from the XPVAL line tests fall into the following categories:

- Packet verification errors.
- Send and receive failures.
- Socket creation failures.
- Checksum errors.
- Miscellaneous errors.

Packet Verification Errors

Packet verification errors indicate problems with either the packet size or the character received. Packet Verification Errors will not abort the XPVAL line tests. Their error messages may scroll off the top of the console terminal screen, preceding a “TCP TEST FAILED” or console message. Packet verification errors are listed below:

MESSAGE: RECEIVE PACKET IS INCORRECT SIZE Expected nn Bytes. Received mm Bytes.

CAUSE: Either message packet was partially lost, or “send” and “receive” are not synchronized.

ACTION: Usually packets will resynchronize with the start of the next segment of the test. However if errors continue for each packet, check surrounding errors, then rerun the test. If problems continue, see

MESSAGE: RECEIVE PACKET NOT VERIFIED First Byte not verified is: xx Should be: y, received: z.

CAUSE: Either byte in packet has changed (bit error) or packets are unsynchronized.

ACTION: Usually packets will resynchronize with the start of the next segment of the test.

However if errors continue for each packet, check surrounding errors, then rerun the test. If problems continue, see Appendix B , “Submitting an CR.”

Send and Receive Failures

Most Send and Receive failures are timing-related. They usually do not abort the tests. Listed below are the Send and Receive failures which do not abort the tests:

Send and Receive Errors

```
TCP MESSAGE RECEIVE FAILED Packet #  {Remote}
IPCSEND FAILED Packet #              {Remote}
DATA RECEIVE FAILED  Packet #        {Remote,
Local}
1ST MASTER SEND FAILED               {Local}
SEND FAILED Packet #                 {Local}
```

Summary Messages:

```
TCP TEST FAILED
LOCAL:  SEND TO REMOTE FAILED
LOCAL:  RECEIVE FROM REMOTE FAILED
LOCAL:  SEND AND RECEIVE FAILED
REMOTE: RECEIVE FROM LOCAL FAILED
REMOTE: END TO LOCAL FAILED
REMOTE: RECEIVE AND SEND FAILED
```

Note the location in the program where the error occurred. For each error, examine the SOCKERR numbers and the Protocol Module numbers returned. Save the error information. Follow the “Actions” for the Protocol Module or NetIPC SOCKERRs, both listed in the *NS 3000/iX Error Messages Reference Manual.*

Socket Creation Failures

Socket creation failures and Network IPC Connection errors cause a test to terminate. Listed below are Socket Errors which abort the tests:

Socket Errors

```
UNABLE TO CREATE SOCKET         {Local & Remote}
CONNECTION REQUEST FAILED       {Remote}
RESPONSE TO CONNECTION FAILED   {Remote}
LOCAL IPCRECVCN FAILED          {Local}
```

Following these errors on the console screen are a SOCKERR and a Protocol Module error. Copy the error messages on the user and system console terminals. Follow the “Action” for the SOCKERR and PM errors, respectively listed in “Network Interprocess Communication Errors” and “Network Transport Protocol Errors” in the *NS 3000/iX Error Messages Reference Manual.*

Checksum Errors

The XPVAL software line tests enable checksum in the TCP protocol of the network transport subsystem. “Checksum” errors may be returned to either console. If “Checksum” errors appear along with “Send and

Receive failures" listed above, then your system may have hardware link problems; see "Investigating the Link" in the *NS 3000/iX Error Messages Reference Manual*.

Miscellaneous Test Errors

Certain errors may appear in all software line tests which do not fit in the categories described above. They are listed here.

SOCKERR #

MESSAGE: PCERRMSG FAILED (SOCKERR #)

CAUSE: Error message could not be acquired from the message catalogue SOCKCAT.NET.SYS.

ACTION: Ensure that the message catalog exists. Examine errors returned to the console before and after this error.

SOCKERR #

MESSAGE: IPCSHUTDOWN FAILED

CAUSE: Socket could not be closed.

ACTION: Examine errors returned to the console before this error. Take action for appropriate SOCKERR.

General Test Suggestions

If the following SOCKERRs appear together, then the network may be "too busy" — that is, coordinating too many processes — to permit proper operation of the XPVAL tests:

Error Message

```
REMOTE ABORTED THE CONNECTION
SOCKET TIMEOUT
```

and

```
CONNECTION REQUEST FAILED
RESPONSE TO CONNECTION FAILED
LOCAL IPCRECVCN FAILED
```

Wait until network activity lessens to execute the tests.

Examine the Protocol Module errors regarding the TCP entity. Protocol Module errors are listed in the "Network Transport Protocol Errors" table in the *NS 3000/iX Error Messages Reference Manual*.

Validate Network Services in Batch Mode

Perform the following steps to use QVALNS to check the Network Services. The services tested are VT, RFA, NFT, RPM, and RDBA. (Note that it is not possible to use passwords with QVALNS. If passwords are required, run NSTEST instead.)

1. Make sure the network transport and Network Services are running on all nodes that are to be a part of this test.
2. Run the NETTOOL utility by entering the program name:

   NETTOOL.NET.SYS

   The root menu will appear.
3. Enter QVALNS to run the Network Services validation in batch mode.
4. When prompted, enter the name of the destination node to which you want to connect. (This is the same as entering the command RUN QVALNS.NET.SYS;INFO=*nodename* outside of NETTOOL.)
5. QVALNS will stream a job that tests the network services. The program will display any errors encountered on the system console.

Validate Network Services Interactively

Perform the following steps to use NSTEST to check the Network Services. It is possible to use passwords with this test.

1. Make sure the network transport and Network Services are running on all nodes that are to be a part of this test.
2. Run the NETTOOL utility by entering the program name:

   NETTOOL.NET.SYS

   The root menu will appear.
3. Enter NSTEST to run the Network Services validation in interactive mode.
4. When prompted, enter the name of the service you want to test. You should always test VT first so that NSTEST can set up a remote session.
5. When prompted, enter the name of the destination node to which you want to connect.
6. When prompted, enter a logon string for the destination node. Enter other values as required. The tool will test the Network Service you selected.

Test RDBA Using NSTEST

To test RDBA, the data base RDBAT must reside in the home group of the remote system. This is not a problem when you run QVALNS, because that program creates the database and then purges it when it finishes. If you want to test RDBA using NSTEST, perform the following steps.

1. Obtain a temporary copy of the job JQVALNS.NET.SYS. If this file is not available, run QVALNS to create it.
2. Find the commands in this job which purge the database. They will be very near the end of the job. Delete these lines using your favorite editor.
3. Stream the job you just edited. When it finishes, the database will be intact so that NSTEST will run.
4. After NSTEST completes, purge the database to save space on your disk.

Stopping Links and Services

You use the NSCONTROL STOP command to stop the Network Services and the NETCONTROL STOP command to stop the links and transport. You should always stop all Network Services before you stop the network transport.

Stop Network Services

NSCONTROL STOP allows all processes that are currently using the services to end normally before the services are actually terminated. If you need to stop the services immediately and do not wish to wait until current processes complete, use the NSCONTROL ABORT command. Be aware that using this command will cause errors to processes that are currently using the services.

Stop a Single Network Service

To stop a specific Network Service or group of Network Services while the remaining active services remain active, enter the NSCONTROL STOP command followed by an equal sign and a list of the services you wish to stop. For example, to stop Remote File Access for both local and remote users, enter the following command:

```
NSCONTROL STOP=RFA,RFAL
```

Stop All Network Services

To stop all Network Services enter the following command:

```
NSCONTROL STOP
```

Current active processes that are using the services will be allowed to complete before the services are terminated.

Abort Network Services

To terminate all Network Services regardless of whether or not currently active processes are using the services, enter the following command:

```
NSCONTROL ABORT
```

Stop Network Interfaces

Stop a Single Network Interface

Issue the following command to stop a single network interface:

```
NETCONTROL STOP;NET=NIname
```

The *NIname* is the network interface name that you configured through NMMGR. Stop other interfaces as required by entering the command using the appropriate NI names.

Stop All Network Interfaces

When you enter the STOP command with no keywords, all entities of the network transport are terminated. (You must terminate the Network Services before stopping the network transport. You will also need to use the DTCCNTRL command if you have an active host-based X.25 interface.) Enter the following command:

```
NETCONTROL STOP
```

Stop a Host-Based X.25 Interface

If the interface you are stopping is a host-based X.25 interface, you must also issue a DTCCNTRL STOP command after you stop the network transport. Enter the commands as shown below:

```
NSCONTROL STOP

NSCONTROL ABORT

NETCONTROL STOP

DTCCNTRL DTC=dtcname;CARD=cardnumber;FUNC=function
```

where function is one of the following:

STOPX25	to stop X.25 services;
STOPPADSUP	to stop PAD support services;
STOPBOTH	to stop both X.25 and PAD support services.

3 Getting Information About the Network

A great deal of information about the network and network connections is available to you through use of various commands and tools provided by the network. This chapter describes the various types of information that you might need access to, describes the tools available for displaying the information, and provides step-by-step instructions for obtaining information where appropriate.

This chapter includes instructions for displaying the following types of information:

- Software version numbers.
- Information about network configuration.
- Status information about link activity, network connections, and Network Services.
- Network performance information (from monitoring network operations).

Verifying Software Versions

Each data communications product consists of a variety of software modules. Each module has an individual version number.

The software modules of all Hewlett-Packard data communications products use a standard version stamp, with the following format:

`v`	The version number of the software. This corresponds to a major revision or a version for a new or revised system environment.
`u`	The update level of the software. This corresponds to a significant revision in product functionality.
`f`	The fix level of the software. This corresponds to a new, supported revision of the software.
`i`	The internal fix level of the software. This is for differentiating special releases of software that do not correspond to a normal release cycle.

A subsystem is a grouping of software modules. The software modules within each subsystem usually have a common or similar function. NS 3000/iX is grouped into the following subsystems:

- The Network Services.
- Network transport.
- Node management services.
- Link support services.
- Node management configurator.

The *vuuff* version stamp fields of the software modules must be the same for all configured software, but the internal fix *iii* fields may differ.

Verify Version of Data Communications Software

Use the NMMAINT utility to display the individual and overall version numbers for all software modules of NS 3000/iX, SNA IMF, and SNA NRJE Network Services, as well as the SNA and NS 3000/iX network link products.

1. Enter the command:

   ```
   RUN NMMAINT.PUB.SYS
   ```

2. If the version, update, and fix levels of these modules do not match, the subsystem will not work correctly. Include the information provided by NMMAINT in any change request (CR) you submit to Hewlett-Packard. See Appendix B , “Submitting an CR,” for information about submitting CRs.

Verify Version of Network Transport Software

To display the version numbers of all of the software modules for the network transport, use the MOD option of the NETCONTROL VERSION command as shown below:

```
NETCONTROL VERSION=MOD
```

Verify Version of Network Services Software

To display the version numbers of all of the software modules for the Network Services, use the MOD option of the NSCONTROL VERSION command as shown:

```
NSCONTROL VERSION=MOD
```

Displaying Configuration Information

You can display information about the location configuration file or network directory file using the CONFIGURATION SUMMARY available as part of NETTOOL. You can also use this tool to compare one configuration file to another.

You can access information about configured network names and addresses using the NAME-ADDRESS MANAGER which is also available through NETTOOL.

Display Local Configuration

Perform the following steps to display a summary of the local system's configuration:

1. Run the NETTOOL utility by entering the program name:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Enter CONFIG to run the CONFIGURATION SUMMARY tool. The CONFIG menu will appear.

3. Enter SUMMARY to display a summary of the network transport and logging configuration values. (Note that logging values may not be recorded for some MPE/iX releases.)

Display Link Configuration Information

You can display the configuration values for a specific link (excluding X.25 links) by using the LINKCONTROL STATUS command. (You can also use the STATUS NETTOOL.)

- To display configuration values for a link on the local node called SYSLINK, enter the following command at the MPE prompt:

  ```
  LINKCONTROL SYSLINK;STATUS=Configuration
  ```

See Chapter 7 , "Commands," for more information on using LINKCONTROL. See Appendix A , "LINKCONTROL Command," for information on the LINKCONTROL status displays.

Display Network Directory Configuration

Perform the following steps to display a summary of the local system's network directory configuration:

1. Run the NETTOOL utility by entering the program name:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Enter CONFIG to run the CONFIGURATION SUMMARY tool. The CONFIG menu will appear.

3. Enter NETDIR to display a summary of the network directory configuration.

Displaying Status Information

Using the various commands and utilities available to you, you can obtain status displays on nearly every aspect of NS 3000/iX. Status displays can help you identify the entities of your network that are currently active or in use. They can also help you diagnose deficiencies in the configured resources that are required for network traffic, such as transmission buffers.

The LINKCONTROL STATUS, NETCONTROL STATUS, and NSCONTROL STATUS commands allow you to display status information about the link, network transport, and services, respectively. The STATUS tool available as part of NETTOOL allows you to display status of the network interfaces and their associated links.

Display Link Status

You can display various levels of status information for a link on the local node using the LINKCONTROL STATUS command.

To display all available status information about a link on the local node called SYSLINK, enter the following command:

```
LINKCONTROL SYSLINK;STATUS=All
```

See Chapter 7 , “Commands,” for more information on using LINKCONTROL. See Appendix A , “LINKCONTROL Command,” for information on the LINKCONTROL status displays.

You can also use NETTOOL STATUS to display link status. See Chapter 6 , “Using NETTOOL,” for information on running NETTOOL STATUS.

Display Network Transport Status

You can display network transport status for a specific network interface or protocol configured on the local node using the NETCONTROL STATUS command. You can also use this command to display only general transport information. In all cases, if the entity is not active, NS 3000/iX will display a warning message telling you that the entity is inactive.

Display General Transport Status

To display the status of the general transport, enter the following command at the MPE prompt:

```
NETCONTROL STATUS
```

Display Status of a NET

To display the status of a NET configured as LAN1, enter the following command at the MPE prompt:

```
NETCONTROL STATUS;NET=LAN1
```

Display Status of an NI

To display the status of an NI configured as LAN1, enter the following command at the MPE prompt:

```
NETCONTROL STATUS;NI=LAN1
```

Display Protocol Status

You can display the status of a general protocol (TCP or PXP), or of a network interface protocol (PROBE, ARP, IP, DIAL or X25) by using the PROT= keyword.

To display the status of the ARP protocol on the LAN1 NI, you would enter the following command at the MPE prompt:

```
NETCONTROL STATUS;NI=LAN1;PROT=ARP
```

Display Network Services

For the Network Services, you can display the status of users, services, or servers.

Display Active Services

To display the services that are currently enabled on the local node, enter the following command at the MPE prompt:

```
NSCONTROL STATUS=SERVICES
```

Display Active Users

To display the number of active users of the Network Services on the local node, enter the following command at the MPE prompt:

```
NSCONTROL STATUS=USERS
```

Display Servers

The servers are the processes that are available to control the network services. You may need to alter the minimum or maximum number of servers available.

To see status information about the available servers, enter the following command at the MPE prompt:

```
NSCONTROL STATUS=SERVERS
```

Displaying Network Performance Information

You can display network performance information using a number of the tools available through NETTOOL. LOOPINIT, RESOURCE MONITOR, and XPPERF all provide performance monitoring features.

Monitor Round Trip Response Time

The LOOPINIT tool allows you to send a series of packets to a specific remote node and display the minimum, maximum, and average times required for the packets to complete the round trip. Chapter 6 , "Using NETTOOL," includes step-by-step instructions for running LOOPINIT.

Monitor Resource Usage

The RESOURCE MONITOR tool allows you to display the current usage of specified resources. This is useful in situations where you suspect over-utilization of a resource.

This tool provides two types of displays: the one-line (non-verbose) display and the detailed (verbose) display. The one-line format lists current use of resources, the maximum experienced (high-water mark), and the maximum allowable usage for specified resources. See the following example.

The verbose mode displays information about a particular item, providing an interpretation of resource usage and pointing to possible relationships with configurable parameters.

Get a One-Line Display

Perform the following steps produce a resource display in non-verbose mode.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the RESOURCE MONITOR tool from the main menu by entering the RESOURCE command. A new menu will appear.

3. Enter DISPLAY to display resource usage. The tool will produce a table similar to the following:

```
THU, MAR 12, 1992,    9:23:39 AM
Item  Subsystem  Name      G/N   Description       Used   High    Max
1     NS XPORT   TCP_CNTL  (G)   Control Buf Pool     0     0    120 :)
2     NS XPORT   CP_POOL_  (G)   Control Buf Pool     0     7    256 :)
3     NS XPORT   1088____  (G)   Control Buf Pool     5    15    240 :)
4     NS XPORT   ROUTER__  (N)   Control Buf Pool     0     7    128 :)
5     NS XPORT   ROUTER__  (N)   Control Buf Pool     0     0     30 :)
6     NS XPORT   ROUTER__  (N)   Control Buf Pool     0     0     20 :)
7     NS XPORT   GPROT     (G)   Control Buf Pool   525   N/A    523 :)
8     NS XPORT   TCP_SIP   (G)   Control Buf Pool   616   N/A    616 :)
9     NS XPORT   UDP       (G)   Control Buf Pool    50   N/A    512 :)
10    NS XPORT   TCP_PM    (G)   Control Buf Pool    33   N/A   2048 :)
11    NS XPORT   ROUTER__  (G)   Control Buf Pool   680   N/A    680 :)
12    NS XPORT   IP_NI     (G)   Control Buf Pool    50   N/A   2048 :)
13    NS XPORT   IP_NI     (G)   Control Buf Pool    50   N/A    512 :)
14    NS XPORT            (G)   Control Buf Pool     1   N/A   1024 :)
15    NS XPORT            (G)   Control Buf Pool     9   N/A    360 :)
16    NS XPORT            (G)   Control Buf Pool     1   N/A    611 :)
[4]RESOURCE>>
```

The column after the maximum allowable usage is an indication of actual usage that may point to over-utilization of a resource. The ":(" indicates normal usage, and the ":)" indicates possible overutilization of a resource. In this example, item 7 shows an indication of overutilization. You could use the verbose mode to obtain a detailed display of that item's resource usage.

Get a Detailed Display

1. While still in the RESOURCE menu, enter detail to toggle the setting of this switch to the detailed mode.
2. Enter ITEM. This will allow you to select the item from the one-line list that you want to display.
3. Enter the item number. For example, you would enter the number 7 to display item #7. RESOURCE MONITOR will produce the detailed display.

Measure Network Performance

You can measure the performance of various aspects of the network, such as the TCP/IP protocol stack, the UDP/IP protocol stack, or X.25 level 3 direct access, using the XPPERF tool. This tool runs by itself or as one of the NETTOOL tools. See Chapter 6 , "Using NETTOOL," for instructions on running this tool using the NETTOOL interface.

Displaying Connection Information

The PING tool allows you to confirm the reachability of a remote node that supports the internet protocol.

You can also use PING to estimate the round trip times before proceeding with lengthy transactions. If you send four or more bytes of data with the echo request, PING displays the round trip times in milliseconds. However, since the echo is performed at layer 3, PING is not the appropriate tool to use when attempting to find out if a particular application is available on the remote node or to estimate application-level round trip times.

You can run PING by itself or as one of the NETTOOL tools. See Chapter 6 , “Using NETTOOL,” for instructions on running this tool using the NETTOOL interface.

Run PING from the Command Line

You can run PING from the command line by using an INFO string. The INFO string must contain the IP address of the remote node and, optionally, the number of packets and number of bytes:

:RUN PING.NET.SYS;INFO=“*ipaddress*[,*packets*][,*bytes*]”

The default number of packets is a continuous stream and the default number of bytes is 64.

Stopping PING

You can enter **[CONTROL]-Y** at any time to exit. The program exits without displaying the menu when run from the command line.

The following examples illustrate using PING with the INFO string. In each case, the parameters echoed by PING are also given.

Within the INFO string, commas are required to separate parameters.

Example 1

This example shows using an INFO string containing all three parameters.

```
:RUN PING.NET.SYS;INFO="15.13.131.59,10,100"

---- PING/iX (ICMP Echo Requestor) : Version X0100003 ----

PARAMETERS INPUT:

Remote IP address in hex  :$0F0D833B
Number of packets         : 10
Number of data bytes      : 100

---- PING $0F0D833B : 100 byte packet(s), 10 packet(s) ----

100 byte(s) from $0F0D833B : icmp_seq =   1, time =   25 ms
100 byte(s) from $0F0D833B : icmp_seq =   2, time =   23 ms
100 byte(s) from $0F0D833B : icmp_seq =   3, time =   24 ms
100 byte(s) from $0F0D833B : icmp_seq =   4, time =   24 ms
100 byte(s) from $0F0D833B : icmp_seq =   5, time =   25 ms
100 byte(s) from $0F0D833B : icmp_seq =   6, time =   24 ms
100 byte(s) from $0F0D833B : icmp_seq =   7, time =   25 ms
100 byte(s) from $0F0D833B : icmp_seq =   8, time =   24 ms
100 byte(s) from $0F0D833B : icmp_seq =   9, time =   26 ms
100 byte(s) from $0F0D833B : icmp_seq =   10, time =  25 ms

---- $0F0D833B PING Statistics ----

10 packet(s) transmitted, 10 packet(s) received, 0 % packet loss
round trip (ms)  min/avg/max = 23 / 25 / 26
```

Example 2

This example shows an INFO string containing the IP address, and five packets. Note that the number of bytes has been defaulted by omitting it in the info string.

```
:RUN PING.NET.SYS;INFO="15.13.131.59,5"

---- PING/iX (ICMP Echo Requestor) : Version X0100003 ----

PARAMETERS INPUT:

Remote IP address in hex : $0F0D833B
Number of packets        : 5
Number of data bytes     : Default of 64 bytes

---- PING $0F0D833B : 64 byte packet(s), 5 packet(s) ----

64 byte(s) from $0F0D833B : icmp_seq =   1, time =   26 ms
64 byte(s) from $0F0D833B : icmp_seq =   2, time =   24 ms
64 byte(s) from $0F0D833B : icmp_seq =   3, time =   23 ms
64 byte(s) from $0F0D833B : icmp_seq =   4, time =   23 ms
64 byte(s) from $0F0D833B : icmp_seq =   5, time =   24 ms

---- $0F0D833B PING Statistics ----

5 packet(s) transmitted, 5 packet(s) received, 0 % packet loss
round trip (ms)  min/avg/max = 23 / 24 / 26
```

Example 3

This example shows an INFO string using the default for number of packets, a continuous stream, of five bytes each. Output is not shown. PING will continue to send data until **[CONTROL]-Y** is entered.

```
:RUN PING.NET.SYS;INFO="15.13.131.59,,5"

---- PING/iX (ICMP Echo Requestor) : Version X0100003 ----

PARAMETERS INPUT:

Remote IP address in hex : $0F0D833B
Number of packets        : Default of continuous stream
Number of data bytes     : 5
```

Error and Information Messages

In addition to the normal reply message details and statistics, PING can display informational and/or error messages. These messages are given below, with an explanation and action to be taken for each message.

User Input Errors (Menu-Driven)

MESSAGE: Invalid IP address. Press RETURN to quit.

CAUSE: An IP address with invalid syntax has been entered for the IP address prompt. The correct syntax for an IP address is *A.B.C.D*—where *A*, *B*, *C*, and *D* are decimal numbers in the range 0–255.

ACTION: Enter an IP address with valid syntax or press **[RETURN]** to quit.

MESSAGE: Invalid number of packets. Press RETURN for default of infinite packets.

CAUSE: An invalid number of packets value has been entered for the number of packets prompt. A valid input is a decimal number in the range 1–65534.

ACTION: Enter a valid number of packets value or press **[RETURN]** to choose the default of sending a continuous stream of packets.

MESSAGE: Invalid number of bytes. Press RETURN for default of 64 bytes.

CAUSE: An invalid number of bytes value has been entered for the number of bytes prompt. A valid input is a decimal number in the range 0–2048.

ACTION: Enter a valid number of bytes value or press **[RETURN]** to choose the default of sending 64 data bytes per packet.

User Input Errors (Command-Line)

MESSAGE: Parameter input error. Quitting...

CAUSE: An irrecoverable error occurred while trying to read the user parameter, either interactively or from the INFO string. This normally happens only when one of the input parameters was out of bounds by an extreme amount.

ACTION: Check the parameters to find the incorrect one and input a valid value.

MESSAGE: Remote IP address is a required parameter.

CAUSE: An IP address was not passed in the INFO string.

ACTION: Pass an IP address as the first parameter within the INFO string.

MESSAGE: Invalid IP address.

CAUSE: An IP address with invalid syntax has been entered in the INFO string. The correct syntax for an IP address is *A.B.C.D* — where *A*, *B*, *C*, and *D* are decimal numbers in the range 0–255.

ACTION: Pass a valid IP address within the INFO string.

MESSAGE: Invalid number of packets. Valid range: 1–65534

CAUSE: An invalid number of packets value has been passed in the INFO string. A valid input is a decimal number in the range 1–65534.

ACTION: Pass a valid value for the number of packets within the INFO string, or omit it to choose the default of sending a continuous stream of packets.

MESSAGE: Invalid number of bytes. Valid range: 0–2048

CAUSE: An invalid number of bytes value has been passed in the INFO string. A valid input is a decimal number in the range 0–2048.

ACTION: Pass a valid value for the number of bytes within the INFO string, or omit it to choose the default of 64 bytes.

Networking Errors

MESSAGE: Receive timeout occurred. Shutting Down...

CAUSE: The PING process has not received any response to its requests for two minutes. It shuts itself down, assuming that the local or the remote side is inactive.

ACTION: This could indicate that the remote node is unreachable. Also check if the local node is congested or hung causing the local ICMP Server not to respond. (The local ICMP Server interacts with PING to send ICMP Echo Requests to the remote and passes incoming replies to the right PING/iX process.)

MESSAGE: Cannot contact local ICMP Server. Shutting down...

CAUSE: PING/iX was not able to contact the local ICMP Server.

ACTION: Check if the transport is active. If not, start the transport.

MESSAGE: Server not accepting requests, as it is busy. Please try later.

CAUSE: Only 15 PING processes can be active at any time. That is, only 15 users can run PING at the same time. Additional users trying to run PING will get this error message.

ACTION: Wait and try later. One of the other PING processes might have completed, allowing you to run the program.

MESSAGE: Cannot resolve path to remote. Path Error, Parm = #Parm_Value. Refer PATH RESULT CODES table in NS 3000/iX Error Messages Manual.

CAUSE: A suitable path out of the local node to reach the remote node could not be found.

ACTION: Look up the table mentioned in the message, under the *Parm_Value* code, and take the action recommended therein.

MESSAGE: Arithmetic trap Parm. Program Quitting.

CAUSE: This is an internal error.

ACTION: Submit an CR with the Parm value, a description of what you were trying to do, and any other abort output that is printed on the terminal.

Internal Errors

The following messages are all internal errors, and should not happen under normal circumstances. In each case, submit an CR.

- `Error opening $STDIN. Program quitting.`
- `Error opening $STDLIST. Program quitting.`
- `Cannot create port. Program quitting.`
- `Internal Error in server. Shutting Down…`
- `Buffer Error in server. Shutting Down…`

Displaying X.25 Information

Several special tools are available to you for use with X.25 network connections. X25CHECK allows you to verify connectivity between two nodes on an X.25 network. X25STAT allows you to monitor the status and statistics for X.25 NIs.

You can run both X25CHECK and X25STAT standalone or from within NETTOOL. Running them from NETTOOL allows you to access help information about the tools. See Chapter 6 , “Using NETTOOL,” for instructions on running these programs from within NETTOOL.

Verify X.25 Connections

Use X25CHECK to create connections to remote X.25 nodes and verify their response. X25CHECK/X25SERVR is actually a pair of programs. X25CHECK runs on the local node which X25SERVR runs on the remote node. The two work together to diagnose conditions between the nodes.

X25CHECK runs at level 3 on the local node. It tries to establish a virtual circuit with the remote node. After the virtual circuit is established, X25CHECK sends the same message to the remote node five times. The program then measures the time period between sending the message and receiving a response from the remote node.

See Chapter 6 , “Using NETTOOL,” for instructions on running X25CHECK.

Monitor X.25 Status

Use X25STAT to monitor status and statistics for X.25 connections. The program will display the contents of the internal X.25 tables, including:

- Global information that is relevant to all connections.
- Socket table information used for level 3 access.
- Connection table information.
- Facilities table information.
- Path table information.

X25STAT displays the information only once. If you want to update the information display you must run the tool again.

Logging and Tracing

Both logging and tracing services are available to you for use as diagnostic and debugging aids.

Logging records subsystem events as selected by the way you have configured logging through NMMGR. Use logging in problem determination and in monitoring network usage and resources.

Tracing is provided at both the user level and at an internal level. User-level tracing provides a record of data communications subsystem intrinsic calls. Internal level tracing records internal state transitions and the sequences of module execution within data communications subsystems. You should only use internal tracing under the recommendation of an HP service representative.

Logging Facility

Node management services, NMS, provides logging services for Network Services, NetIPC, network transport, and all data communications links. Logging is performed at three levels: network logging, event logging, and link level logging. Network logging records the usage of the communications network resources. It serves as a tool in resolving network problems. Event logging records the major subsystem events. The `NSCONTROL` command with the `LOG=` option can be used to enable or disable detailed event logging for the Network Services (see Chapter 7 , “Commands,” for more information). The link level logs to MPE/iX log files only.

You can configure logging to record messages to the console, to a log file, or to both for each individual subsystem. See the *HP e3000/iX Network Planning and Configuration Guide* or the *NS 3000/iX NMMGR Screens Reference Manual* for information on how to configure logging.

Three commands are available to help you manage log files. `SHOWNMLOG` displays the name of the current log file and shows the space that is still available in the file. `SWITCHNMLOG` allows you to close the current log file before it is full and open a new one. `RESUMENMLOG` allows you to reactivate logging after a recoverable error. See Chapter 7 , “Commands,” for information on these commands.

Tracing Facility

Tracing is provided for the Network Services subsystem, Network Interprocess Communication (NetIPC), the network transport subsystem, and the link subsystems. You enable tracing for the Network Services by the DSLINE command for each user's services. Network Service tracing is used to trace messages generated by your applications. For more information, see *Using NS 3000/iX Network Services*.

You enable tracing for NetIPC applications with the NetIPC intrinsic IPCCONTROL, which is explained in the *NetIPC 3000/XL Programmer's Reference Manual*.

You can selectively enable tracing for the network transport with the NETCONTROL command see Chapter 7 , "Commands." You can enable tracing at the link level in the NMMGR configuration for some links, as explained in the *NS 3000/iX Screens Reference Manual*. You can also enable link level tracing with the LINKCONTROL command.

Trace Files

Network transport trace records are written to disk files and are of file type NTRAC. Trace files are named either by explicitly specifying a file name (in the configuration file or with the NETCONTROL command) or by using the default trace file filename. If you explicitly specify a file name, the contents of the file are overwritten each time a new trace is started. No warning is issued. If you use the default file name, NMS uses NMTC*nnnn*.PUB.SYS as a file name. In the file name, *nnnn* is a number from 0000 to 9999. Each time a new trace is started, NMS opens a new file and increments *nnnn* by one, thus creating a new file name. If this new trace file name is the name of a file that already exists, NMS continues to increment *nnnn* by one until it produces the name of a new (non-existing) file. If the NMS trace facility reaches an end-of-file mark while recording to a disk file, it wraps subsequent entries around to the beginning of the file and overwrites the previous entries.

Format Log and Trace Files

You can format log and trace files into a readable format using the NMDUMP utility. You can run NMDUMP by itself or as one of the NETTOOL tools. See Chapter 6 , "Using NETTOOL," for step-by-step instructions for running NMDUMP using NETTOOL.

NMDUMP allows you to select specific subsystems and message types for formatting. (Note that you must have configured logging so that messages of the type you select are recorded.) For example, you may only need to see critical error messages for a LAN link. NMDUMP will let you select just these messages to be formatted. See *Using the Node Management Services (NMS) Utilities* for a table of the logging subsystems and message types that you can select for formatting.

Format X.25 Log Files

Messages for X.25 links are not recorded to the same logging file as messages for other links. If you need to format log messages for a host-based X.25 link, you should see *Configuring and Managing Host-Based X.25 Links* for information on using the EVLOG formatter. If you are logging messages for an X.25 link with PC-based network management, see *Using the OpenView DTC Manager* for information.

4 Troubleshooting Process

Troubleshooting data communications problems can be a very involved process since there are many hardware and software components to investigate. You will be able to quickly identify and resolve some problems, however. These include invalid software installation, version incompatibilities, insufficient MPE/iX resources, corrupt configuration files, programming or command errors, and file system errors.

Other problems will require more investigation. Once you identify the problem, it is likely that you will be able to resolve the problem using the suggestions in this chapter or the detailed instructions provided in the *NS 3000/iX Error Messages Reference Manual*.

This chapter includes information on the following topics:

- How to identify problems.
- Characterizing problems.
- Identifying possible causes of problems.

Once you have identified the problem and the possible cause, use the strategies described in Chapter 5 , “Common Network Problems,” to further isolate and correct the problem.

Identifying Problems

The usual method of identifying problems is to characterize the situation in which the problem occurs and then investigate which of the possible causes are actually responsible for the problem. Finding the cause is often sufficient to suggest the resolution of the problem. For example, assume that the problem is characterized as "the user is unable to open a line with the `DSLINE` command." A possible cause is that the user entered a command using incorrect syntax. You would resolve the problem by correcting the command and reissuing it. However, if the syntax was correct, you would have to look for another possible cause, such as an inactive link or a failure of the remote node.

Thus, in most cases you start with the characterization of the problem and investigate the possible causes. The difficult part of troubleshooting is to identify the actual cause of the problem. Once you know the actual cause, you can take the appropriate action to resolve the problem.

Characterize the Problem

It is important to ask questions when you are trying to characterize a problem. Start with global questions and gradually get more specific. Depending on the response, you ask another series of question, until you have enough information to understand exactly what happened.

Key questions to ask are as follows:

1. Was an error message generated? Use the *NS 3000/iX Error Messages Reference Manual* to look up the cause of the error and take the action suggested. If this does not resolve the problem, continue with the next question.
2. Is the problem isolated to one user or program? If so, continue to the next question. If more than one user is involved, proceed to question 6.
3. Did the user perform the operation correctly? Was syntax correct? Does the user have the correct logon and authority to use the command or service? Correct any problems found. If the operation was correct, continue with the next question.
4. Did the problem occur while the user was running a program? Were there program errors? If so, investigate and correct the program errors. Otherwise, continue with the next question.
5. Did the problem occur while attempting to open a line or transmit data? If so, investigate the connection between this system and the remote system.

6. If more than one user is involved, does the problem affect all users? The entire node? If so, has anything changed recently? Some possibilities are:

   - New software and hardware installation.
   - Same hardware but changes to the software. Has the configuration file been modified? Has the MPE/iX configuration been changed?
   - Same software but changes to the hardware.

7. Do you suspect hardware or software?

   It is often difficult to determine whether the problem is hardware or software related. Symptoms that mean you should suspect the hardware are:

   - Bad LAN card or PSI dumps.
   - Link level errors, either returned to the user or logged to the console. This includes CI errors, NMERR errors, power fails, and link shutdowns.
   - Lost data—data is sent but not received at the link destination. (This could also be caused by a software problem.)

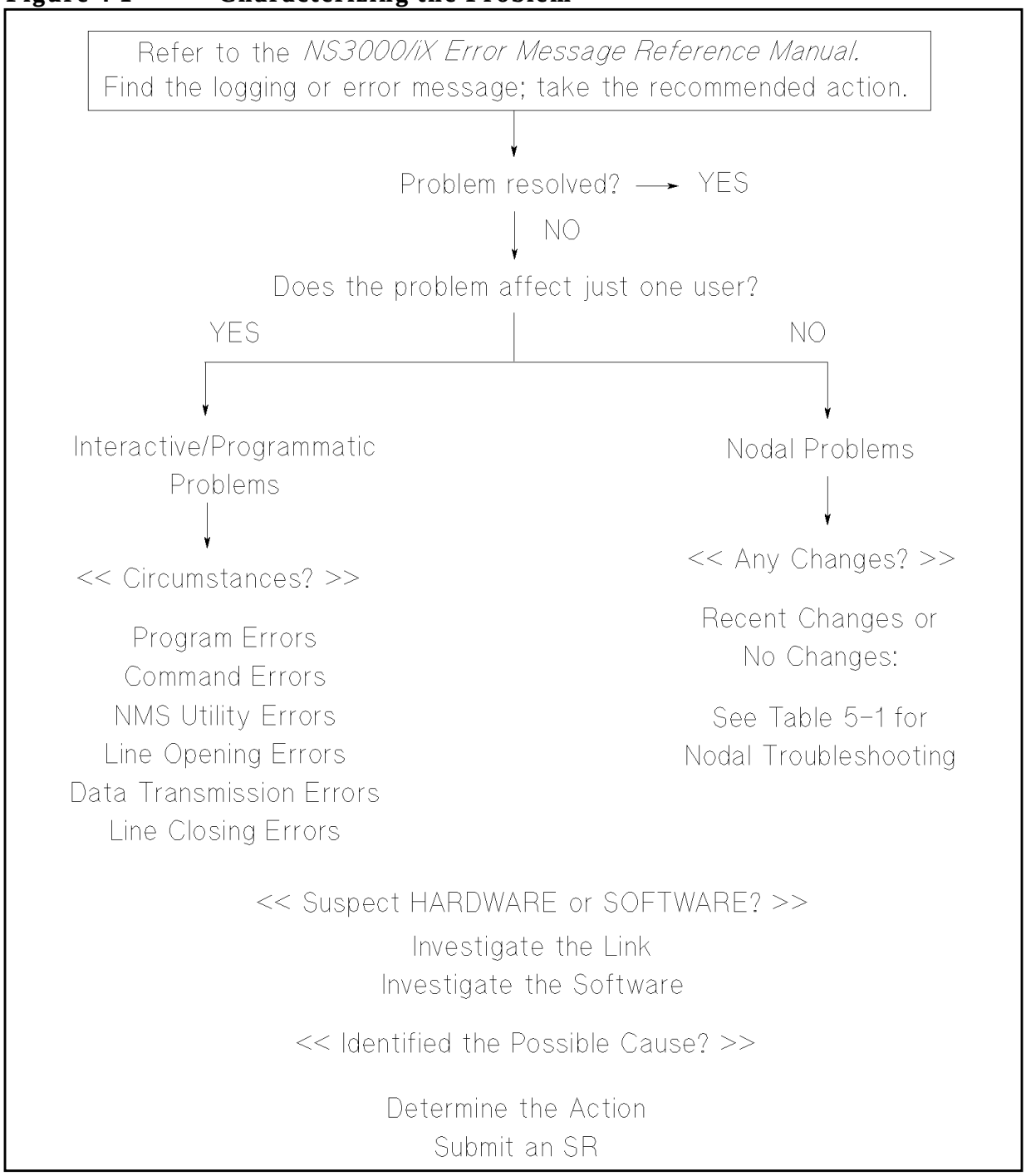
   Symptoms that mean you should suspect the software are:

   - Logging messages at the console.
   - Network Services errors returned to users or programs.
   - MPE/iX file system (FSERR) or command interface (CIERR) errors (except “Remote Not Responding” errors).
   - Data corruption.
   - Terminal hangs.
   - Intermittent errors.
   - Network-wide problems.

Identify the Cause of Problems

The type of investigation that you use to identify the possible causes of a problem depends on whether the problem affects one user or an individual situation, or if the problem is node-wide. Once you have the answers to the questions listed previously, use the flowchart in Figure 4-1 as a guide and see Chapter 5 , “Common Network Problems,” for a problem resolution strategy.

Figure 4-1 Characterizing the Problem



5 Common Network Problems

This chapter presents strategies for dealing with some of the more common network problems. Once you have characterized a problem and identified a possible cause using the troubleshooting guidelines provided in Chapter 4 , “Troubleshooting Process,” find the problem in this chapter and follow the strategy presented here to resolve the problem.

This chapter provides strategies for dealing with the following types of network problems:

- Interactive or programmatic problems.
- Command errors.
- Nodal problems.
- Link problems.
- Software problems.

Interactive or Programmatic Problems

The first step in investigating interactive or programmatic problems is to examine any error message returned. If you have received a specific error message, find it in the *NS 3000/iX Error Messages Reference Manual* and take the action recommended. Most error messages are easily understood by the user or programmer, although some of the explanations refer to internal procedures comprehensible only to qualified HP representatives. Users are not expected to understand these explanations, but they should take the actions listed.

Program Errors

If the user is using any of the programmatic capabilities of NS 3000/iX and an intrinsic completes with an error, the recovery procedure depends upon the intrinsic. How you check for the error code depends on which service you are using.

- If a NetIPC intrinsic was issued that received a condition code indicating an error, use the IPCCHECK intrinsic to obtain additional details. Always check for the PM error code; this is essential to identify the cause if the network transport is unable to complete a request.
- If a file system intrinsic was issued that received a condition code indicating an I/O error occurred, use the FCHECK intrinsic to obtain additional details.

Command Errors

If you are using the interactive capabilities of NS 3000/iX and associated links and receive an error, refer to "NS 3000/iX Network Services Error Messages" in the *NS 3000/iX Error Messages Reference Manual*.

The command errors fit into four categories:

- **Syntax errors or invalid options**. These errors result from user errors when issuing the command. They are readily corrected by checking for the correct syntax and reissuing the command.
- **Warnings**. If a command is executable but may give unexpected results, a warning is issued. This would occur in a situation where conflicting options were specified. The warning informs you which option was actually used (or not used).
- **Resource Errors**. These errors occur when a system resource needed for the execution of the command is not available. If they occur, you can wait and reissue the command later, when the resource may be available. If resource errors happen frequently the configuration or resource allocation of the system may be inadequate. The network manager may need to investigate further.
- **Internal Errors**. These errors indicate that the software is malfunctioning. If they ever occur, notify your HP representative. The network manager should follow the steps outlined in Appendix B , "Submitting an CR.""

For syntax errors and warnings, consult the reference pages in this manual for that command's correct syntax and options, or refer to *Using NS 3000/iX Network Services*.

Line Opening Errors

There are several reasons why a `DSLINE` command for opening a communications line might be rejected. Some line opening errors actually occur when a `REMOTE HELLO` (or `DSCOPY`, or programmatic `RPMCREATE` or `FOPEN`) is executed, not when the `DSLINE` is done. The following list summarizes the likely causes of line opening failures:

- The user made a syntax error in the `DSLINE` command.
- The user specified an erroneous *`nodename`* or *`envid`* in the `DSLINE` command. The node name must match the one configured for the system the user is trying to reach. Make sure that all users know the correct node names. You may want to post a map with the configured node names for all the nodes on the network. The correct node names can be checked in the network directory (if one is being used).

- A network was not started by the local console operator, the remote console operator, or any intermediate nodes. Check that all required commands have been issued on the local, remote, and intermediate nodes. The network interfaces, the LAN NI, the loopback NI, point-to-point NI, X.25 NI, token ring NI, and gateway half NI, must be initialized with NETCONTROL START commands. The DTC/X.25 Network Access card in the DTC must be started; refer to *Using the OpenView DTC Manager* or *Configuring and Managing Host-Based X.25 Links*. The Network Services must be initialized with the NSCONTROL START command. Some links may also need to be started by NETCONTROL ADDLINK commands. Links can be configured not to be started when a NETCONTROL START command is executed and be started via NETCONTROL ADDLINK. Links may have been closed by NETCONTROL DELLINK or may have been closed because an irrecoverable error was detected on the line.
- The remote node may not be operational.
- The remote operator may have lowered the session limit. This would cause a failure in a REMOTE HELLO or a DSCOPY or RPMCREATE that tried to automatically log on to a session.
- The local console operator may have used the service list of the NSCONTROL command to limit the Network Services to incoming users only. On the remote node, the operator may have limited the Network Services to outgoing only. This would cause a REMOTE HELLO, DSCOPY, remote FOPEN, or RPMCREATE to fail, depending on which services were not started.
- If the line is a dial up line, a failure in a REMOTE HELLO, DSCOPY or RPMCREATE can be caused by the following:
  - — If auto dial, the number was busy, wrong, or was never answered at the remote computer.
  - — The security strings did not match at either the local or remote node (if security was enabled).
  - — The IP address of the remote node was not configured as a candidate for use of this link.
  - — If the link is a shared dialup link, a failure will occur if the link is connected to a node different than the one issued in the DSLINE command.
  - — When a REMOTE HELLO is issued which causes the phone to be dialed, there is a window in which subsequent REMOTE HELLOs from other users will be rejected. The window is from the time the auto dial starts (or dial request) to when the connection is established.
- A REMOTE HELLO, DSCOPY or RPMCREATE will fail if the IP address of the remote node configured in the network directory does not match

the IP address of the remote node configured in the NS Configuration file.

- All virtual terminals on the remote node are already in use, which means there are no remote resources available to establish a remote session. This would cause a failure in a `REMOTE HELLO` or a `DSCOPY` or `RPMCREATE` that tried to automatically log on to a session.
- Someone has exclusive access to the specified line or the user requested exclusive access to a line that is already in use.
- Someone is exclusively accessing a server program. For example, someone is executing the `STORE` command or a `SYSGEN` system backup on `DSSERVER.NET.SYS`.
- There is a hardware problem—the communications device is not responding correctly.

Line Closing Errors

There are several reasons why a `DSLINE` command for closing a communications line might be rejected. The following list summarizes the likely causes of line closing failures:

- The user made a syntax error in the `DSLINE` command.
- The user specified an erroneous *`nodename`* or *`envid`* in the `DSLINE` command. The nodename must match the one configured with NMMGR. Make sure that all users know the correct nodenames. You may want to post a map with the configured nodenames for all the nodes on the network.
- The remote node may not be operational.
- There is a hardware problem — the communications device is not responding correctly.

NMS Utility Errors

A file system error (FSERR) may have occurred while attempting to access the configuration file. Try to access the configuration file under the same user ID using NMMGR. Use the NMMGR Error screen to find out what the underlying FSERR is. A complete listing of NMS error messages is available in *Using the Node Management Services (NMS) Utilities*. Correct the problem and retry.

Nodal Problems

The first step in investigating nodal problems is to examine any error message returned. Error messages returned by NS 3000/iX and associated links are listed in the *NS 3000/iX Error Messages Reference Manual*, along with their meaning and recommended recovery action.

If you have received a specific error message, find it in the manual and take the action recommended. Only if there is no clear error or the recommended action does not correct the problem is it necessary to investigate further. Follow the strategy shown in Figure 4-1.

Recent Changes

If you begin experiencing problems immediately following either a new installation or changes to the software or hardware, often you can easily identify what is causing the problem. Table 5-1 shows the symptoms and possible causes for a new installation, changed software and changed hardware, respectively. This table also suggests a course of action for situations where no recent changes have been made.

Once you have identified the possible cause, you may need to isolate the actual cause. Proceed to Investigate the Software or Investigate the Link, depending on the nature of the possible cause. For more information on some of the possible causes, including what to do when you have isolated the actual cause, proceed to Determine the Action. If the recommended action is to contact an HP representative, use the guidelines in Appendix A of the *NS 3000/iX Error Messages Reference Manual*.

Table 5-1 Nodal Troubleshooting Strategy

Changes	Symptom	Possible Causes
New installation	Console locked or hung;serious failures. System abort	Software installation invalid. Configuration incorrect, serious internal error.
Software changes	System abort DSCOPY command aborts.	Configuration incorrect, serious internal error. NFTCAT.NET.SYS is bad; incompatible version, or MAKECAT was not done.
Hardware changes	Unable to use NS.	For LAN, LAN card not properly connected to MAU or network cable, (LOOP, twisted pair) or cable not properly installed (missing or bad terminator, twisted pair not connected to hub). For Fiber Distributed Data Interface/iX, either the Media Interface Connector (MIC) receptacle is not properly connected to the FDDI device adapter or the FDDI concentrator. For NS 3000/iX Point-to-Point, either the PSI card is not properly connected to the cable, or the cable is improperly installed (missing or bad terminator). Also check all modem or other connections. For DTC/X.25 iX Network Links, the DTC/X.25 Network Access card may not be properly installed in the DTC.
No changes	Unable to use services or a warning that old services are being used. Cannot connect to remote system.	NSCONTROL has been stopped or network has been shut down, or NSCONTROL has been issued to limit the number of active servers. See “Line Opening Errors.”

Investigate the Link

The following is a strategy to use to identify and solve link problems. You should use this strategy if you are not sure what is causing the problem because many times errors in the upper level software are due to hardware problems. You can also use this strategy if you have identified a hardware-related possible cause and need to isolate the actual cause.

LAN, Token Ring, FDDI, 100VG-AnyLAN, 100Base-T Link Problems

For problems that involve LAN, Token Ring, FDDI, 100VG-AnyLAN, or 100Base-T link, use the following strategy where applicable:

- Issue the `LINKCONTROL` *`linkname`*`; STATUS=DIAGSTATS` command. Inspect the output and attempt to identify the problem. Refer to the *Online Diagnostic Subsystem Manual, Volume I,* for a detailed analysis of the fields displayed. Retain a copy of the output from this command for your Hewlett-Packard representative.
- Run `PING` to confirm whether or not the remote node is reachable. See Chapter 6 , “Using NETTOOL,” for instructions on running `PING`.
- If `PING` fails, use the LAN node diagnostic that is appropriate for the type of card on your system:

Card	Online Diagnostic
LAN	LANDAD
	LAN3PBB
	CONSOLAN
Token Ring	LAN5PBB
FDDI	FDDIPBA
100VG-AnyLAN	VGPBA
100Base-T	VGPBA

  Refer to your hardware documentation for information on these diagnostics. These diagnostics are online tools that verify the hardware components by running the self-test, then a series of tests of the cables and connectors.

- If a failure has taken place, give the files `NMLG`*`xx`*`.PUB.SYS` and `NETDMP`*`nn`*`.PUB.SYS` to your Hewlett-Packard representative for additional analysis.

If the problem is easily reproducible, and link level tracing was inactive when the problem took place, turn on tracing using the LINKCONTROL command. When the problem has been reproduced, turn off trace and give this trace file to your Hewlett-Packard representative for additional analysis. If a hardware failure takes place while trace is active, give the files NMLG*xx*.PUB.SYS and NETDMP*nn*.PUB.SYS to your HP representative as well.

The log message contains an error code, such as an NMERR. Information on the cause and recovery of these errors can be found in the *NS 3000/iX Error Messages Reference Manual*. Keep a copy of the log file and the output. If you need to submit an CR, send the log file and output to your Hewlett-Packard representative.

- If link level logging is not enabled, enable it through NMMGR so that the information will be available if this problem can be repeated.

NS Point-to-Point 3000/iX Link Problems

The NS Point-to-Point 3000/iX link (router link) is connected with a programmable serial interface (PSI) card. For problems that involve the PSI, use the following strategy where applicable:

- Issue the LINKCONTROL *linkname*; STATUS=DIAGSTATS command. Inspect the output and attempt to identify the problem. Refer to Appendix A , "LINKCONTROL Command," for a detailed analysis of the fields displayed. Retain a copy of the output from this command for your Hewlett-Packard representative.
- Run PING to confirm whether or not the remote node is reachable. See Chapter 6 , "Using NETTOOL," for instructions on running PING.
- If PING fails, use PSIDAD. PSIDAD is an on-line diagnostic tool. It verifies the PSI components by running the PSI self-test, then extends the testing as far into the communications network as possible, depending on which equipment is connected to the PSI. Refer to the *On-Line Diagnostic Subsystem Manual, Volume I,* for instructions.
- If a PSI failure has taken place, give the files NMLG*xx*.PUB.SYS and NETDMP*nn*.PUB.SYS to your Hewlett-Packard representative for additional analysis.
- If the problem is easily reproducible, and link level tracing was inactive when the problem took place, turn on tracing using the LINKCONTROL command. When the problem has been reproduced, turn off tracing. Save both the raw trace file and the formatted output for your Hewlett-Packard representative for analysis. It is important to save any PSI dump file (NETDMP*nn*.PUB.SYS) that is created while link level tracing was enabled. Send both the PSI

dump file and the link trace file to your Hewlett-Packard representative for additional analysis.

- Check the MPE/iX log file for I/O error logging. Format the log file. Keep a copy of the file and the output for your Hewlett-Packard representative to study.

NOTE

If you lose connections on an NS 3000/iX Point-to-Point link as a result of successive power failures, you can recover the connections by issuing the following commands:

`NETCONTROL NET=`*`niName`*`;DELLINK=`*`linkName`*

`NETCONTROL NET=`*`niName`*`;ADDLINK=`*`linkName`*

DTC/X.25 iX Network Link Problems

The DTC/X.25 iX Network Link operates using a DTC/X.25 Network Access card on the DTC. For problems that involve the DTC, perform the following steps when applicable:

- Issue the `LINKCONTROL` *`linkname`*`; STATUS=DIAGSTATS` command on the LAN link, where *`linkname`* is the name of the DTS link. Inspect the output and attempt to identify the problem. Retain a copy of the output from this command for your Hewlett-Packard representative.
- Use the OpenView DTC Manager to verify the status of the DTC/X.25 Network Access card if you are using PC-based network management. Use `TermDSM` to verify the status of the DTC/X.25 Network Access card if you are using host-based network management.

Investigate the Software

Follow the strategy described below to identify and solve any problems that might involve software.

- There may be version incompatibilities between different software subsystems. This is essential to check for if new software has recently been installed on your node. Use the software verification utility `NMMAINT` to display the version identification numbers of the software modules. Compare the first five characters of these version IDs with those listed as compatible with each other in the System Status Bulletin, Software Release Bulletin, `N00N` files or other HP source. If a discrepancy is found, locate a known set of compatible software and install it.
- Issue the `LINKCONTROL STATUS` command. Inspect the output and attempt to identify the problem. Refer to Appendix A , "LINKCONTROL Command," for a detailed analysis of the fields displayed. Retain a copy of the output from this command for your Hewlett-Packard representative.
- Check the configuration file. Use NMMGR to print the data screens. Inspect the output and attempt to identify the problem. Follow the suggestions provided in the section "Corrupt Configuration Files" later in this section. Retain a copy of the output for your Hewlett-Packard representative.
- In general, the log files are the best source of information. They should be checked for any problem encountered. Use the command `SWITCHNMLOG` to isolate the specific log file immediately after the problem occurs. Use the time range option of NMDUMP whenever possible to further narrow the focus on when the problem occurred. Inspect the formatted output and attempt to identify the problem. Retain a copy of the output from the log file for your HP representative.
- If the cause of the problem cannot be isolated with any other means, or if the recommended action has not resolved a problem, then use the line tests described in this manual. The intent is to verify each component of the hardware and software individually in hopes of isolating the faulty component. Inspect the output and attempt to identify the problem. Retain a copy of the output from these tests for your HP representative.
- If the problem is easily repeated and NMS tracing was inactive when the problem took place, turn on tracing using the `NETCONTROL TRACE` command. When the problem has been reproduced, turn off tracing and give this trace file to your HP representative for additional analysis.

- If the problem causes a system failure, take a full memory dump of the system. Format the system dump with the Dump Analysis Tool (DAT) and send the formatted tape to your HP representative.

Common Problems and Actions

Invalid Software Installation

A software installation may be invalid. Run NMMAINT.PUB.SYS to obtain a listing of version IDs for NMS and for all of the NMS dependent subsystems.

Locate the overall version IDs for each subsystem. Check that these subsystems are the correct version for operation with the associated link.

MPE/iX Configuration Incorrect

Refer to *System Startup, Configuration, and Shutdown* to obtain an I/O listing of the system. Check that the drivers are correctly configured.

Insufficient MPE/iX Resources

There may be insufficient MPE/iX resources, such as configured table sizes. Refer to the recommendations for system tables provided in *System Startup, Configuration, and Shutdown*. Reconfigure MPE/iX to fix any problems found and restart the system.

Corrupt Configuration File

The configuration file is possibly corrupt. If the error persists, use NMMGR to manually check the configuration file (if possible). Check to see that all data records have been created. If bad records seem to be localized to a particular item, delete that item and reconfigure it. If necessary, RESTORE a known good backup copy of the file.

Corrupt Network Directory File

If the network directory file is open in NMMGR during a system failure, starting the network transport with NETCONTROL START does not recover the network directory file. Run NMMGR in maintenance mode as follows:

```
:FILE NMMGRCMD=$STDINX
:RUN NMMGR.PUB.SYS
NM Configuration Manager 32098-20012 A.02.00 (C) Hewlett Packard Co.  1986
NMMGROPENDIR NSDIR.NET.SYS
NETWORK DIRECTORY: Recovering file NSDIR.NET.SYS
NMMGREXIT
```

After recovering the file, stop and restart the network transport as described in Chapter 2 , “Operating Your Network,” of this manual.

Incompatible Configuration File Version

Run the NMMGRVER.PUB.SYS program to convert the old configuration file to the new format. Refer to the *Using the NMS Utilities* manual for more information.

Insufficient Configuration File Values

Only change the configured values in the configuration file for a persistent or widespread problem. The configured values apply to communication over all the connections and with all the remote nodes in the internet. The default values are calculated to provide good performance in a variety of situations. Changes to these values may improve one situation but affect other situations adversely. If the recommended action for a particular error or log message is to change the configured value, do so only for an extremely high number of log messages or for repeated error messages. Consult your HP representative for more information.

Retransmission Timeout Errors

The network transport provides reliable end-to-end communication. As part of ensuring reliable receipt of packets, the transport protocol TCP keeps track of the packets transmitted. If TCP does not receive an acknowledgment within the configured time period, TCP retransmits the packet. If the packet is retransmitted the maximum number of times configured and is still unacknowledged, then TCP logs a retransmission timeout error and aborts the connection.

The transport protocol PXP may also log a retransmission timeout error. This occurs in much the same way as described for TCP, although PXP retransmits requests, not packets, and waits for replies, not acknowledgments. PXP is only used whenever an IPCLOOKUP is issued as part of a NetIPC application, and only communicates with the socket registry.

Retransmission timeouts can occur for the following reasons:

- Packets were transmitted to a remote node which was not active or which terminated before the packet arrived.
- Excessive node loads took place during connection establishment.
- The remote node experienced congestion or lack of buffers.
- Possible link or configuration problems.

If a retransmission error is returned in a log message or in an IPCCHECK error code for NetIPC applications, first check that the remote node is up and that its transport has been started. If so, check if the retransmission timeout error is an isolated event or an ongoing problem. Examine a formatted log file for the period up to and including

the error.

If the problem is ongoing, then take the appropriate action:

- If the log messages show initial TCP connection failures due to a heavily loaded remote node, configure a longer Initial Retransmission Interval for the Transmission Control Protocol (TCP) Configuration Screen. This is in the NETXPORT branch of the NMMGR network configuration. Also, you can increase the connection assurance interval on this screen if there are a large number of TCP connections configured.
- If there are IPCLOOKUP failures, and the log messages show PXP timeouts due to a heavily loaded remote node, configure a longer default retransmission interval for the PXP data screen. This is also in the NETXPORT branch.
- If the problem affects established connections and none of the above conditions apply, then configure a longer retransmission interval upper bound, or configure a higher number of maximum retransmissions per packet for the Transmission Control Protocol (TCP) Configuration Screen.

NetIPC Errors

NetIPC programmatically creates processes on both local and remote systems. These processes must be released, along with any descriptors and resources, after the program completes. Unless these process are terminated properly, errors may result.

NetIPC Shutdown Errors

The NetIPC call IPCSHUTDOWN releases a descriptor and any resources that are associated with it. Since system resources are used as long as call sockets and destination sockets exist, it is important that application programs release the sockets whenever they are no longer needed.

Before a process terminates, it should terminate its connection with IPCSHUTDOWN. Because this termination takes effect very quickly, all of the data that is in transit on the connection is lost when the connection is shut down. As a result, the processes that share a connection must cooperate to ensure that no data is lost. Indications of a faulty shutdown procedure on an individual or application level are:

- If you receive log messages or NetIPC error codes where the recommended action for some of the log messages is to increase the number of TCP connections, and the connections are not currently active.
- If the TCP PM log message indicates that a packet was received after the IPCSHUTDOWN call but before the TCP connection was fully deleted.

Indication of a faulty shutdown procedure on a nodal level is an incomplete shutdown of the network transport.

Network Transport Shutdown

Shutting down the network transport via the `NETCONTROL STOP` command requires that all NetIPC call sockets, all TCP connections, and all PXP sockets are closed. An error (Transport Shutting) is returned to all open sockets. Until this error is received by the user and the reply sent to TCP/PXP by NetIPC, the network transport does not terminate. The Network Services shut down completely even if an `NSCONTROL ABORT` has not been issued. However, it is important that user applications always have a send or receive posted on any open socket so that the shutdown error is delivered to them.

The only way to tell if the network transport has completely shut down is to check the log file for the Control Process; Transport Stopped and the TCP SIP/ General Protocol Stop nodal log messages. If these messages have not been logged, the network transport is waiting for an open socket and cannot completely terminate. The network transport may be re-initialized even though the “old” transport has not completely terminated. The two versions do not interfere with each other, and the old one goes away when its last open socket is finally closed. This old transport does not use any CPU and does not retain “ownership” of the links, but the data structures that wait on the open connection do use virtual memory.

If you find any of these indications, check any NetIPC applications for a faulty shutdown procedure. Refer to the *NetIPC 3000/XL Programmer's Reference Manual*.

6 Using NETTOOL

The NETTOOL utility allows you to run a complete set of networking diagnostic programs under a common user interface. It provides help information on its commands and on its core functions. It also allows you to add your own applications, so that you can perform diagnostic operations common to your site while taking advantage of the NETTOOL user interface and facilities.

This chapter describes NETTOOL and its operation. It includes information on the following topics:

- Tools available in NETTOOL.
- How to use NETTOOL.
- How to use each of the NETTOOL tools.
- How to add your own tools to NETTOOL.

NETTOOL Tools

Each of the NETTOOL tools provides a specific functionality that will assist you in troubleshooting network problems, monitoring resources, or simply accessing information about your network and its operations.

Types of Tools

There are three types of tools that run in the NETTOOL environment. Core tools and Associated tools are provided by Hewlett-Packard and are always available to you. User tools are applications that you develop at your local site but choose to attach to the NETTOOL utility for ease of use and convenience. Instructions for attaching user tools are included later in this chapter.

Core Tools

Core tools are those tools that are an integral part of the NETTOOL package. Core tools run only as part of NETTOOL and have a consistent user interface. While running a core tool, you can use any of the NETTOOL commands and access the NETTOOL help system. Help is available on all aspects of the core tools. All core tools recognize **[CONTROL]-Y** inputs.

Associated Tools

Associated tools are tools that were developed as standalone programs but that Hewlett-Packard has attached to the NETTOOL environment for convenience. These are programs that have proven useful for one purpose or another.

Because associated tools were developed independently of NETTOOL, they may have a different look and feel from the core tools. While you are running an associated tool from within NETTOOL, you will not be able to access the NETTOOL commands or help information.

User Provided Tools

User provided tools are programs that you have developed locally but wish to include as part of the NETTOOL environment. You can provide your own help information for user tools and access the information using the NETTOOL facility. However, NETTOOL help information will not be available from within a user tool.

A major advantage to attaching user tools to NETTOOL is that you are then able to run all such tools from one point in the MPE file system.

Differences

Table 6-1 summarizes the differences between core tools, associated tools, and user tools.

Table 6-1 Differences in Tool Types

	Core	Associated	User
Consistent user interface	Yes	No	No
Access NETTOOL help within tool	Yes	No	No
Use NETTOOL global commands	Yes	No	No
Control-Y recognized	Yes	Tool dependent	Tool dependent
Run standalone	No	Yes	Yes
HP factory support	Yes	Yes	No

Available Tools

Table 6-2 summarizes the tools available in NETTOOL, the type of each tool, and the function of each tool.

Table 6-2 The NETTOOL Tools

Tool	Type	Function
CONFIGURATION SUMMARY	Core	Displays a summary of the information in the configuration or directory file.
filters	Core	Displays global filter setup.
IPCINT	Associated	Provides a command interface to IPC.
LOOPINIT	Associated	Monitors round-trip response time between nodes.
NAME-ADDRESS MANAGER	Core	Displays local cache of node names and addresses.
NMDUMP	Associated	Provides formatting and analysis capabilities for system dumps.
NSTEST	Associated	Interactively provides a quick validation of the Network Services.
NSLOGON	Associated	Establishes temporary connections between nodes to quickly validate the network transport.
PING	Core	Allows the local system to send a message to one or more remote nodes and examine their response.
QVALNS	Associated	Provides a quick validation of the Network Services. Runs in program mode.
RESOURCE MONITOR	Core	Displays the internal resources for the network transport.
SOCKINFO	Associated	Displays socket information.
STATUS	Core	Displays the status of the network interfaces and the associated links.
X25CHECK	Associated	Creates connections to remote X.25 nodes and verifies their response.
X25STAT	Associated	Monitors the status and statistics for X.25 network interfaces. Also displays internal data structures.
XPPERF	Associated	Provides a cursory performance measurement.
XPVAL	Associated	Provides a quick validation of the network transport.

Using NETTOOL

You can run NETTOOL either interactively or through a batch job. In interactive mode, you can take advantage of the flexibility provided by the NETTOOL menu structure. You can also access the available help information.

If you need to perform a simple operation, however, you might choose to run in program mode, passing the information required to run the tool you have chosen in the run command.

Running NETTOOL Interactively

To run NETTOOL interactively, perform the following steps.

1. Enter the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Enter the command from the main menu that corresponds to the tool that you want to use. NETTOOL will launch the tool you select.

3. For instructions on running a tool, see the section that describes that tool.

NOTE

NETTOOL can also accept an info string from the :RUN command. For example, you could enter the following command to run the NAME-ADDRESS MANAGER to display the name cache:

```
:RUN NETTOOL.NET.SYS;INFO="NAMEADDR;CACHE;NAME;QUIT"
```

Getting Help

Help is available on any command that is valid at the current point in the NETTOOL command tree. It is not possible, however, to direct the output of a help request using the OUTFILE command.

- To see a list of the commands available at the current menu, type:

  ```
  HELP (or ?)
  ```

- To see a list of the available commands with a one line description, type:

  ```
  HELP COMMANDS
  ```

- To see a description of a specific NETTOOL command, type:

  ```
  HELP commandname
  ```

  If the same command is available in several menus, the help

information you see will be the information that pertains to the way the command operates for the current menu. Abbreviations are not allowed for the command name.

- To get general information on NETTOOL use, type:

  ```
  HELP OPERATION
  ```

- To get help on adding comments to input scripts, type:

  ```
  HELP COMMENT
  ```

- To get help on executing MPE commands from within NETTOOL (the colon capability), type:

  ```
  HELP COLON
  ```

- To browse the entire help text, type:

  ```
  HELP BROWSE
  ```

Using Commands

To execute a command within NETTOOL, you enter the command and any parameters required for the desired execution of the command. Each menu provides a list of the commands that are available at that point in the NETTOOL utility. Commands may be abbreviated.

Commands may be chained together, up to 150 characters total. Separate the individual commands by semicolons. For example:

```
NAMEADDR;CACHE;OUTFILE myfile;FILTERS
```

If any command requires the inclusion of a semicolon, then that command must be the last one in a command string.

Global Commands

A number of commands are available from all NETTOOL menus. These commands are listed below along with their function.

DATA	Enable/disable NETTOOL data going to $STDLIST. If you have not defined an output file with an OUTFILE command, data will go to $STDLIST regardless of this setting. If you have defined an output file and the data flag is on, data goes to both $STDLIST and the data file. If you have defined an output file and the data flag is off, data goes only to the data file. The data flag is on by default. See also MESSAGES command.
DEBUG	Enter the MPE debug facility. Return to NETTOOL when done.
DO	Execute a specific command from the redo stack, using the syntax DO *commandnumber*. If no *commandnumber* is

	specified, execute the previous command.
EXIT	Step back one level in the command menus. If you enter this command from the root level, NETTOOL will prompt you to determine if you really meant to quit.
HELP	See a list of commands available.
HELP ALL	View all the help text for the current menu.
HELP BROWSE	Browse through the entire help file for all of NETTOOL.
HELP COMMANDS	See a list of commands with one line descriptions.
HELP *command*	Get detailed help on the specified command.
INFILE	Redirect NETTOOL input commands from the file specified, using the syntax INFILE *filename*[.*group*[.*account*]]. NETTOOL will read and execute commands from this file until all commands are executed or an error is encountered in the command input.
LISTREDO	Show a list of previously executed commands (the redo stack).
MAIN	Return directly to the root menu.
MANUAL	Format the NETTOOL manual into a file for printing.
MENUS	Turn on/off the available commands display. Command displays are on by default.
MESSAGES	Enable/disable NETTOOL messages going to an OUTFILE. If the messages flag is on, messages will go to the file defined in the OUTFILE command, if the file exists. If the messages flag is off, messages will go only to $STDLIST. It is not possible to prevent messages from going to $STDLIST. The messages flag is on by default. See also DATA command.
OUTFILE	Redirect NETTOOL output to the file specified, using the syntax OUTFILE *filename*[.*group*[.*account*]]. The redirection will remain in effect until you issue a new command to specify a different output file, cancel redirection by entering the OUTFILE RESET command, or exit NETTOOL.
QUIT	Exit NETTOOL from any point in the menus.
REDO	Make changes to last command and then execute the command again. You can choose a command from the redo stack by specifying its command number using the syntax REDO *commandnumber*. The “d”, “i”, and “r” edit commands are allowed as well as direct replacement.

SETVAR	Set variable to given value using the syntax SETVAR *variable value.*
SHOWVARS	Show the variables in use.
VERSION	Display the revision numbers of NETTOOL modules and of the NS transport.
:	Interactively execute MPE commands.
:MPEcommand	Execute one or more MPE commands then return to NETTOOL.
#	Designates a comment. Comments may be inserted in a command string or in an INFILE record. Comments must be separated from any previous text in a record or string by a semicolon.
?	Shows the current menu.

Running NETTOOL in Batch Mode

You can run NETTOOL from a job with very few restrictions. The commands can come from either a job file or from an input file. For example, you could use the following job to print a copy of the NETTOOL manual in batch mode:

```
!job nettool,user/userpass.account/acctpass;outclass=pp,2
!nettool.net.sys
:file printdev;dev=pp;env=elite2.hpenv.sys
manual
*printdev
60
quit
!eoj
```

You could perform the same operation using an INFILE for the user input, as follows:

```
!job nettool,user/userpass.account/acctpass;outclass=pp,2
!nettool.net.sys;info="infile filename"
!eoj
```

The input file must contain all the user input, including the QUIT command.

Keep the following in mind when running NETTOOL in batch mode:

- Be sure to consider any optional parameters. For example, if an output file for a command might already exist, you will need to tell the program whether or not to purge it.
- Help is not available on the commands or user tools if an INFILE is active.

Using the NETTOOL Tools

The following sections describe each of the available tools and provide information on their use. You can access additional information from within NETTOOL by asking for help on the tool from the main menu.

NOTE You can use abbreviations for the NETTOOL commands. The abbreviations must uniquely identify the command at the current menu.

CONFIGURATION SUMMARY

The CONFIGURATION SUMMARY tool provides options that let you display information from the network configuration and network directory files. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   NETTOOL.NET.SYS

   The root menu will appear.

2. Select the CONFIGURATION SUMMARY tool from the main menu by entering the CONFIG command. A new menu will appear.

3. Select the menu item or items needed to display the information you want to see. The available choices are described as follows.

SUMMARY — Select this command to generate a summary of the configuration file, NMCONFIG.PUB.SYS. Optionally, you can specify a different configuration file using the syntax:

SUMMARY confilename

You can also specify a different file using the conffile command.

NETDIR — Select this command to generate a summary of the network directory file, NSDIR.NET.SYS.

COMPARE — Select this command to compare the contents of two configuration files. You can specify the files to use in the command, using the syntax:

COMPARE *altfile conffile*

If you do not specify an altfile, the program will prompt you for one. If you do not specify a conffile, the program will use NMCONFIG.PUB.SYS.

You can limit the comparison to just a subset of records using the subtree option.

`filters`	This option displays the current values of the global filters, `conffile`, `altfile`, and `subtree`, as well as the current settings of the global filters.
`conffile`	Use this option to select a configuration file for the `SUMMARY` and `COMPARE` options.
`altfile`	Use this option to select an alternate file for the `COMPARE` option.
`subtree`	Allows you to specify a subset of records to be compared by the `COMPARE` option. For example, if you specify `NETXPORT.NI.LAN1`, the program will check only those screens in the file whose name starts with `NETXPORT.NI.LAN1`. To set this value back to the default (root), press **[RETURN]** at the `subtree` prompt.

Filters

The filters tool displays global filter setup.

1. Run `NETTOOL` by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the filters tool from the main menu by entering the filters command. Filter options will be displayed as follows:

INFILE:	default	none
OUTFILE:	default	none
MESSAGES FLAG:	default	none
DATA FLAG:	default	none
MENUS FLAG:	default	none
NODE NAME FILTER:	default	none
IP address:	default	none
GFLAGS	default	none

NOTE

GFLAGS is a toggle key. It could be “SET” or “NOT SET” by typing “GF”.

If GFLAGS is “SET” then global and local filters will be the same.

If GFLAGS is “NOT SET” then only the local filter will change and local will take priority over the global filter.

IPCINT

The IPCINT tool provides a command interface to IPC. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the IPCINT tool from the main menu.
3. Enter a NetIPC intrinsic abbreviation. You will be prompted for any parameters required by the intrinsic.
4. To exit the tool, type ex at the prompt.

IPCINT creates a log file, IPCLOG, to track its actions.

LOOPINIT

The LOOPINIT tool sends a series of packets to a specific remote node and monitors the round-trip response time. It displays the minimum time, maximum time, and the average time. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the LOOPINIT tool from the main menu.
3. You will be prompted for a remote node name. Enter the name of the node that you want the test packets sent to. If you wish, you may enter the local node name.
4. You will be prompted to specify information on frame text or for frame length, if you do not specify frame text. Enter values as required.
5. You will be prompted for the number of frames to be sent. Enter the number desired.

LOOPINIT will display the minimum, maximum, and average times, in milliseconds, required for the frames to make the round trip. It will also allow you to display a histogram which graphically represents the times. If you choose not to display the histogram, simply enter an N at the prompt.

NAME-ADDRESS MANAGER

The NAME-ADDRESS MANAGER tool provides options that let you display the local cache of node names and addresses. This tool is useful in detecting duplicate IP addresses and permits you to clear entries in the name cache if necessary. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   NETTOOL.NET.SYS

   The root menu will appear.

2. Select the NAME-ADDRESS MANAGER tool from the main menu by entering the NAMEADDR command. A new menu will appear.

3. Select the menu item or items needed to display the information you want to see. The available choices are described as follows.

CACHE — Select this item to display or delete information stored in the name and address cache. A new menu will appear presenting you with the choices as described:

- NAME — Select to display name cache entries as specified by nodefilter. If looking for duplicate IP addresses, set nodefilter to @. (If the filter is not set, it displays all names.)
- DELNAME — Select to delete a name entry from cache. Syntax is DELPATH *nodename*. This is useful in case of a duplicate name in the name cache.
- DELPATH — Select to delete a name entry from cache plus IP address mapping. Syntax is DELPATH *nodename*.
- LOCAL — Select to display local node name.
- TOTALS — Select to display total number of names in cache and total number of names in directory.
- filters — Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag).
- nodefilter — Select to set the name filter (@, #, and ? wildcards are allowed).
- ipfilter — Select to set the IP address filter. Enter a single address or @ for all. Enter the IP address as four positive integers between 0 and 255 separated by

periods or blanks (for example, 15.123.44.98).

MAPPINGS — Select this item to obtain information about the correspondence between IP addresses and link addresses or to delete mappings from the table. A new menu will appear presenting you with the choices as described:

- MAPPING — Select to display all mappings between IP address and link address for those entries selected by subnetfilter and addrfilter.
- DELMAPPING — Select to delete mapping information of IP address to network address. Syntax is DELMAPPING ipaddress. For example,

  DELMAPPING 15.13.128.1
- TOTALS — Select to display total number of mappings.
- filters — Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag)
- subnetfilter — Select to set the subnet filter. Specify the name used in the NETCONTROL command.
- addrfilter — Select to set the address filter. You will be prompted for the address type. Enter IP, ETHER802, X25, or NONE as required.
- sorting — Select to specify the sorting method for the output of the MAPPING option. You will be prompted for the sort type. Enter IP or LINKADDR as desired.

ROUTING — Select this item to obtain information about the gateways used to access different subnets. A new menu will appear presenting you with the choices as described:

- ROUTING — Select to display routing information as specified by the networkfilter and gatewayfilter settings.
- DELROUTING — Select to delete specified routing.
- GATELIST — List all started DCNs, (gateways).

	GATE UP	A specific gateway (Active).
	GATE DOWN	A specific gateway (Not Active).
	STATISTICS	Display the statistics; IP Route statistics, IP update routing table etc.
	TOTALS	Select to display total number of routings.
	filters	Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag).
	networkfilter	Select to set the networkfilter. Enter a single IP address or @ for all. Enter the IP address as four positive integers between 0 and 255 separated by periods (for example, 15.123.44.98).
	gatewayfilter	Select to set the gatewayfilter. Enter a single gateway IP address or @ for all. Enter the IP address as four positive integers between 0 and 255 separated by periods (for example, 15.123.44.98).
PATH		Select to obtain information about the different addresses or names used at different layers in order to access a remote destination. You will be prompted to specify the type of information you need. Enter NAME or ADDRESS as desired.
	NAME	Select to display addresses at different levels.
	ADDRESS	Select IP address to get corresponding path information for that IP address.
	filters	Select to see current global filter settings. Displays the settings of INFILE, OUTFILE, messages flag, data flag, and menus flag.

NMDUMP

NMDUMP is one of the node management services (NMS) utilities. You use this tool to decode and format log records or trace messages so that they can be more easily read and analyzed.

NOTE You cannot use NMDUMP to format X.25 log or trace files. For information on X.25 logging and tracing, refer to *Using the OpenView DTC Manager* for PC-based systems or to *Configuring and Managing Host-Based X.25 Links* for host-based systems.

Perform the following steps to format records from the current log file.

1. At the MPE prompt, enter the SHOWNMLOG command to obtain the name of the current log file. Record this name. You will need to enter the name of the file you want to format when you run NMDUMP.
2. At the MPE prompt, enter the SWITCHNMLOG command to close the current log file and begin recording log and trace information to a new log file.
3. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.
4. Select the formatting tool from the main menu by entering the NMDUMP command. The NMDUMP menu will appear.
5. Select the menu options that will allow you to specify the type of records to format (log or trace).
6. Select additional menu options as required to specify the exact information you want to format.
7. When prompted for the name of a file to format, enter the file name you recorded in step 1. You will also be prompted to enter a name for the output file. The default output file is $STDLIST.
8. To exit NMDUMP at any time, enter // at any prompt.

See *Using the Node Management Services (NMS) Utilities* for more information on the options available in NMDUMP.

NSTEST

The NSTEST tool allows you to test the Network Services interactively. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

The root menu will appear.

2. Select the NSTEST tool from the main menu.
3. When prompted, enter the name of the service you want to test. You should always test VT first so that NSTEST can set up a remote session.
4. When prompted, enter the name of the destination node to which you want to connect.
5. When prompted, enter a logon string for the destination node. Enter other values as required. The tool will test the Network Service you selected.
6. Test other services as required.

NSLOGON

The NSLOGON tool allows you quickly verify that the network transport is operating correctly. It uses the NetIPC intrinsics to establish a connection to a well-known server on a remote node. Therefore, both the network transport and the Network Services must be started on all nodes before you use this tool. You can choose whether to contact all nodes or selected nodes by responding to the NSLOGON prompts. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the NSLOGON tool from the main menu.
3. You will be prompted to specify whether or not you want to logon to all nodes in the directory. Answer yes or press **[RETURN]** to logon to all nodes, otherwise answer no.
4. Respond to additional prompts as required.
5. NSLOGON will produce a list of node names along with an indication of whether or not the logon to each node was successful.

PING

The PING tool allows you to test remote connections by sending messages to one or more remote nodes and examining their response. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the PING tool from the main menu by entering the PING command. A new menu will appear.
3. Select the menu item or items needed to perform the PING requests you want to perform. The available choices are described here.

PING	This option sends ICMP echo requests to remote systems. On receiving the ICMP echo replies, the program displays the number of packets sent and received and the time that it took each packet to complete the round trip. You can specify the destination by name or by IP address. If you specify by name, you can choose a single node or a set of nodes by using wildcards (@, #, and ?). If you specify by address, the ping will go to that specific address. Enter the IP address as four positive integers between 0 and 255 separated by periods or blanks (for example, 15.123.44.98).
RANGEPING	Use to send ping messages to a range of IP addresses. The syntax is RANGEPING *lowerip higherip*. Enter the IP address as four positive integers between 0 and 255 separated by periods (for example, 15.123.44.98). If you do not enter the boundary IP addresses the program will prompt for them.
GATEPING	Use to send ping messages to each of the existing gateways in the nodes routing table.
filters	Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag).
number	Use to specify the number of packets the program sends for each request. The range is from 1 to 1,000,000. The default is 5.
size	Use to specify the size of the packets the program sends for each request. The range is from 8 to 2,048 bytes. The default is 64.
nodefilter	Use to select multiple nodes to be acted on by subsequent PING requests (@, #, and ? wildcards are allowed).
ipfilter	Use to select a remote IP address to be acted on by subsequent PING requests. Enter the IP address as four positive integers between 0 and 255 separated by periods or blanks (for example, 15.123.44.98). Standalone PING requires periods.

QVALNS

The QVALNS tool allows you to test the Network Services in program mode. To use this tool, perform the following steps.

1. Make sure the network transport and Network Services are running on all nodes that are to be a part of this test.
2. Run the NETTOOL utility by entering the program name:

   NETTOOL.NET.SYS

   The root menu will appear.
3. Enter QVALNS to run the Network Services validation in batch mode.
4. When prompted, enter the name of the destination node to which you want to connect. (This is the same as entering the command RUN QVALNS.NET.SYS;INFO=*nodename* outside of NETTOOL.)
5. QVALNS will stream a job that tests the network services. The program will display any errors encountered on the system console.

RESOURCE MONITOR

The RESOURCE MONITOR tool provides options that let you display resource usage according to the current settings of the resource filters. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   NETTOOL.NET.SYS

   The root menu will appear.
2. Select the RESOURCE MONITOR tool from the main menu by entering the RESOURCE command. A new menu will appear.
3. Select the menu item or items needed to display the resource information you want to see. The available choices are described here.

DISPLAY	Use to display resource usage for the resources specified by the type parameter.
CLEAR	Use to set the high-water mark values for a chosen item to zero.
RESET	Use to reset all resource filter values to their defaults.
filters	Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag).
detail	Use to toggle between detailed (verbose) and one-line (non-verbose) modes. Verbose mode displays information about a particular item detailing

interpretation of resource usage and pointing to possible relationships with configurable parameters. Non-verbose mode displays current, maximum experienced (high-water mark), and maximum allowable usage for the resources specified. Default is non-verbose.

`item` Use to select a particular item from the one-line display so that you can obtain detailed information on that item.

`refresh` Use to set the number of times the program will display resource usage before returning control to you. Default is one cycle. Selecting **[CONTROL]-Y** will also return control.

`type` Use to select which resource types the program will display in the one-line (non-verbose) mode. Default is to display all resource types.

`used` Use to suppress display of entries that are currently unused. Default is to display resources regardless of usage.

`delay` Use to select the interval (in seconds) between displays of resource usage. Use this option in conjunction with `refresh` in order to monitor the activity of resource usage. Default is a delay of 1 second.

SOCKINFO

The `SOCKINFO` tool displays sockets information.

1. Run `NETTOOL` by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the `SOCKINFO` tool from the main menu by entering the following `SOCKINFO` syntax:

   ```
   SOCKINFO [filter][,pin]
   ```

   `filter` program name in the form: `file.group.account`; or user name in the form: `user.account`.

   `pin` display for the specified PIN instead of starting out in Global Display.

3. If filters are not used, SOCKINFO will print a Global Display like the following:

```
GLOBAL DISPLAY    Host=sampsys              Gsxds=$a.d5690000  10:30 am
------------------------------------------------------------------------
  Pin  User                    Program                  Job     Pri   Skts
  59  (system process)        snmp.net.sys                      cq152    2
  61  (system process)        sockreg.net.sys                   lq149    1
  63  (system process)        dsdad.net.sys                     lq149   14
  69  joe.mpe                 vtserver.net.sys         s538     lq100    1
  70  bob.mpeix               vtserver.net.sys         s546     lq100    1
  79  spool,unispool.sys      system3.unispool.sys     j138     de208    2
  80  spool,unispool.sys      system6.unispool.sys     j138     de202    0
  81  spool,unispool.sys      system6.unispool.sys     j138     de202    1
  82  spool,unispool.sys      system3.unispool.sys     j138     de206    0
         : etc...
 447  veruser.nmpascal        vtserver.net.sys         s570     lq152    1
------------------------------------------------------------------------
Totals: 153 processes, including 1 locked semaphore; 177 sockets.
```

4. Select the options needed to display the information you want to see by typing one of the single characters as shown here:

?	Print help text.
:	Enter MPE command mode.
A	For an interpreted and raw dump of a socket data structure. (PM capability required, must be in Process Display mode)
C	List all open call sockets and datagram sockets.
D	Call HPDEBUG. (PM capability required)
E	Exit this program.
F	Define Global Display filters.
G	Enter Global Display mode.
H	Print a history of processes displayed.
I	List configured IP addresses.
L	Display locked LSI semaphore entries. (PM capability required)
M	Toggle display of internet address/host name in Socket Display.
O	Toggle display of object addresses, enter Global Display.
P	Enter Process Display mode.
Q	Enable/disable semaphore queuing. (Default is to

	not queue)
R	Enter Destination Display mode. (Must be in Process mode)
S	For an interpreted dump of a socket data structure. (Must be in Process Display mode)
T	Enable/disable tracing.
V	Print the SOCKINFO version number.
Y	Define new timeout value, in seconds. (Default is 0 : disabled)

5. To return to NETTOOL, type E.

STATUS

The STATUS tool provides options that let you display the status of the network interfaces and their associated links. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the STATUS tool from the main menu by entering the STATUS command. A new menu will appear.

3. Select the menu item or items needed to display the information you want to see. The available choices are described here.

NODE	Select to display the local node name, the domain name if one is configured, the CPU type, the MPE version, the transport version, and the transport start time. The display information is from the configuration file (not the name cache displayed when you use the NAME-ADDRESS tool).
INTERFACES	Use to obtain a list of all the network interfaces and the links configured for those NIs. (You can obtain additional information about the links using the LINKCONTROL command or the X25STAT tool. You can obtain additional information about NIs using the NETCONTROL command.)
TCPSTAT	Use to display TCP global statistics and connection table information. Available commands for the TCPSTAT menu are TCPGLOBAL, CONNTABLE, and CONNINFORMATION.

NOTE CONNINFORMATION — function not available at this time.

IPSTAT	Use to display IP statistics for the network specified by niname. If you have not set niname, you will see statistics for all NIs.
LKSTAT	Use to display statistics for the link whose name has been set by lkname. If you have not set lkname you will see statistics for all links. (The statistics shown will be the same as those displayed by the LINKCONTROL command.)
PROBESTAT	Use to display probe statistics for inbound and outbound packets for the network specified by niname. If you have not set niname, you will see statistics for all NIs.
ARPSTAT	Use to display ARP statistics for the network specified by the niname command. If you have not set niname, you will see statistics for all NIs.
UDPSTAT	Use to display global UDP statistics or to report UDP sockets statistics information for the network specified by niname. If you have not set niname, you will see statistics for all NIs.
filters	Select to see current filter settings for this menu. Also displays the global settings (INFILE, OUTFILE, messages flag, data flag, and menus flag)
niname	Use to set the name of the network interface for the ARPSTAT, IPSTAT, PROBESTAT, TCPSTAT, and UDPSTAT commands to act upon. The default is @ (display statistics for all NIs).
lkname	Use to set the name of the link for the LKSTAT command to act upon.
detail	Use this toggle to specify the level of detail that the program will display. If this filter is set, the program will display full statistics for the link. If it is not set, the program will display only summary statistics.
refresh	Use to set the number of times the program will display statistics before returning control to you. Default is one cycle. Selecting **[CONTROL]-Y** will also return control.
delay	Use to set the number of seconds which will be inserted as a delay after each statistics display. If the refresh filter is set to a value of 1, the delay filter has no effect. If you enter the delay command and press [RETURN], the default value of 1 second is set. Note that the delay time is in addition to any processing time for the program. That is, setting a delay of 1 does not guarantee that the statistics

measurements will occur at one second intervals. You should view this parameter as a means of causing successive measurements to be space by *at least* the delay time.

`recent` Use this filter to select whether the displayed statistics will be adjusted to show only the data which occurred recently. If the flag is not set (the default), the program will display *all* statistics totals.

X25CHECK

The X25CHECK tool creates connections to remote X.25 nodes and verifies their response. It also provides information that allows estimation of the performance of the network and its load. The remote node runs a background program, X25SERVR, that responds to X25CHECK. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the X25CHECK tool from the main menu.
3. You will be prompted for a remote node name and network name. Enter the name of the node and network that you want the test packets sent to. If you wish, you may enter the local node name.
4. X25CHECK will set up a VC to the remote node and send ten messages. The remote node will echo the messages back. At the end of the test, the program clears the connection but keeps the server running so that you can set up a connection to if different node if you desire.
5. To terminate the server, use **[BREAK]** and ABORT or ABORTJOB.

X25STAT

The X25STAT tool monitors the status and statistics for X.25 network interfaces It displays internal data structures. To use this tool, perform the following steps.

1. Run NETTOOL by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the X25STAT tool from the main menu.
3. You will be prompted to enter either a table specification or a counter specification; both cannot be entered on the same command line. (If an NI name is not entered on the command line, X25STAT will

display only the started X.25 NI.)

4. The program will display the contents of the internal X.25 tables. The information prints only once. To get new, updated information, you will need to run X25STAT again.

XPPERF

The XPPERF tool measures the performance of the TCP/IP protocol stack, the UDP/IP stack, or X.25 level 3 direct access. The program interfaces to the transport through the IPC intrinsics. You must run XPPERF on both the local system and a remote system for the test to work, with the program on the remote system started first. To use this tool, perform the following steps.

1. Have someone at the remote location run NETTOOL on the remote system by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. The remote user should select the XPPERF tool from the main menu.
3. The remote user will be prompted for the protocol, the mode (master/slave), and other test values. The user must specify `slave` as the mode. The remote user should set other values as agreed upon.
4. Run NETTOOL on the local system by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

5. Select the XPPERF tool from the main menu.
6. Respond to the prompts as required, or press **[RETURN]** to accept defaults. For the local system, you must specify `master` as the mode.
7. XPPERF will write the measured data to a file named XPERFD in the local group.

XPVAL

The XPVAL tool provides a quick validation of the transport by setting up a connection between two nodes. You must run XPVAL on both the local system and a remote system for the test to work, with the program on the local system started first. To use this tool, perform the following steps.

1. Run NETTOOL on the local system by entering the program name at the MPE prompt:

   ```
   NETTOOL.NET.SYS
   ```

   The root menu will appear.

2. Select the XPVAL tool from the main menu.

3. XPVAL will prompt you for information it needs to run the validation. Respond as required.

4. Make sure XPVAL is running on the remote node as well and have the remote user enter information about the remote node.

5. XPVAL will run a one minute connection test to verify the operation of the transport and report any errors it encounters.

   See Chapter 2 , “Operating Your Network,” for a list of XPVAL line test error messages.

Adding Your Own Tools

You can add you own diagnostic tools to by following some simple rules. You can also provide help information on the user-provided tools.

Add User Tools

Information needed to run a user-provided tool must reside in USERINFO.NET.SYS. You can create this file with any text editor. It must have a record length of 80 characters or less.

You can describe up to 20 user tools in the file. For each tool, two types of entries are needed: the one word command which will initiate the tool from the root level of NETTOOL, and a list of the MPE commands required to start the tool (as if it were being used standalone).

The first character in the tool command must be alphabetic. The command can be up to 20 characters in length. It must not duplicate any NETTOOL global command, core tool name, or associated tool name.

The first character in a line containing an MPE command must be an exclamation mark. Characters may be either lower or upper case, but NETTOOL does not distinguish between the two. NETTOOL does not interpret the string after the exclamation mark in any way. Up to 79 characters may follow the exclamation mark.

The list of MPE commands must follow the command name. For example, a valid set of entries in the file might be:

```
NEATPROG2
!file input=fromhere.pub.sys
!file output=tohere.net.sys
!run myprog.maui.hawaii;info="map, 26, verbose";lib=p
TESTTOOL
!run testtool.net.sys
```

To run MYPROG.MAUI.HAWAII from the NETTOOL root, a user would enter the command NEATPROG2. To run TESTTOOL.NET.SYS, the user would enter TESTTOOL.

Any first character in a line other than !, a..z, or A..Z is an error and will cause all subsequent entries to be ignored. Also, if you specify more than five MPE commands for a tool, all subsequent entries in the file will be ignored. Blank lines have no effect. Lines with only a ! will be sent to MPE as a carriage return.

A sample USERINFO file is included as part of the NETTOOL package.

Add User Provided Help

To provide help on user tools, you must create the file USERHELP.NET.SYS. This file should contain help text for all user tools defined in USERINFO.net.sys.

Use the following format for help text:

1. There must be an \ENTRY=ROOT block which gives a one word list of the user-defined NETTOOL commands. These commands are the same as those defined in USERINFO.NET.SYS. This is the text that will be displayed when the user types help with no parameters at root level.
2. Within the \ENTRY=ROOT block, there must be an \item=commands block that contains a one line description of the user-defined commands that run the tools. This is the text that will be displayed when the user types help commands at the root level.
3. Within the \ENTRY=ROOT block, there must be an \ITEM=*command_name* block for each of the tools. These blocks contain the text that is displayed when the user types help *command_name* at the root level where *command_name* is a user command defined in USERINFO.NET.SYS.
4. There must be an \ALL directive at the end of the help text.

A sample USERhelp file is included as part of the NETTOOL package.

For a user-defined tool defined as follows:

```
mytool
!file input=parms.net.sys
!run myprog.net.sys
```

The user help might look like this:

```
\ENTRY=ROOT   MYTOOL
  MYTOOL
\ITEM=COMMANDS
  MYTOOLExamines the path cache and purges all entries
\ITEM=MYTOOL
  If you suspect that the path cache is out of unused
  entries or that duplicate IP addresses have been
  defined, use MYTOOL to clear the entire cache.
\ALL
```

1. You must format the help file into a help catalog using MAKECAT.PUB.SYS, as follows:

```
file input=sourcefilename
run makecat.pub.sys,help
rename helpcat,userhelp.net.sys
reset input
```

Here, *sourcefilename* is your unformatted help file and helpcat is the file name reserved by MAKECAT for its output.

7 Commands

This section describes the NS 3000/iX network commands for the NS 3000/iX services and associated links. The commands are listed in alphabetical order and described in Table 7-1.

NOTE You must have NM capability to execute any of the following commands.

Table 7-1 NS 3000/iX Network Commands

Command	Description
LINKCONTROL	Provides link information, or activates or deactivates link level tracing.
LINKCONTROL STATUS	Requests status information about the link.
LINKCONTROL TRACE	Activates or deactivates link level tracing.
NETCONTROL	Initiates, terminates, and controls the operation of the network transport.
NETCONTROL ADDLINK	Dynamically adds a configured network link to the active network configuration.
NETCONTROL DELLINK	Dynamically deletes a configured network link from the active network configuration.
NETCONTROL START	Initiates the network transport functional entities.
NETCONTROL STATUS	Displays the status of the network transport functional entities.
NETCONTROL STOP	Terminates the network transport functional entities. Immediately terminates the Network Services. (You should always terminate Network Services via NSCONTROL commands first.)
NETCONTROL TRACE	Enables or disables message tracing for a specified network transport functional entity.
NETCONTROL UPDATE	Dynamically updates selected network transport configuration parameters for an active network interface.
NETCONTROL VERSION	Displays the version of the software modules of the network transport.
NSCONTROL	Initiates, terminates and controls the operation of the Network Services.
NSCONTROL ABORT	Immediately terminates the Network Services.
NSCONTROL AUTOLOGON	Enables or disables the autologon feature for the NFT, RFA, and RPM remote network services.
NSCONTROL LOADKEYS	Loads the Network Services command keywords. Used for localization.

Table 7-1 NS 3000/iX Network Commands

Command	Description
NSCONTROL LOG	Enables or disables detailed logging (configured as CLAS0004 of SUB0006) for the Network Services.
NSCONTROL SERVER	Alters the characteristics of the Network Services server processes.
NSCONTROL START	Initiates the Network Services.
NSCONTROL STATUS	Displays status information about the Network Services.
NSCONTROL STOP	Allows existing users to continue with current task, but prevents initiation of any new tasks or new users for the Network Services.
NSCONTROL VERSION	Displays the version of the software modules of the Network Services.
RESUMENMLOG	Resumes logging after a recoverable error.
SHOWNMLOG	Displays the identification number and available space of the log file.
SWITCHNMLOG	Closes the current log file and creates and opens a new one.

LINKCONTROL

Activates or deactivates link level tracing on a specified communications link. Provides link transmission error statistics and/or configuration information.

Syntax

```
                                [A(ll)          ]
                                [C(onfiguration)]
LINKCONTROL linkname  {;STATUS=} [L(inkstate)    ]
                      {,       } [S(tat)istics)) ]
                                [D(iag(stats))  ]
                                [R(eset)        ]

                                [,DATA   ]
                           [ON] [,ALL    ]
LINKCONTROL Linkname;TRACE=[OFF][,PARTIAL][,buffsize][,tracefile]
                                [,FULL   ]
```

Use

```
Available          In Session?          YES
                   In Job?              YES
                   In Break?            YES
                   Programmatically?    YES
Breakable?                              YES
Capabilities?                           NM
```

Parameters

linkname The configured name of an active data communications link. Only the link name specified in the LINK portion of the configuration file (@LINK.*linkname*) is valid. The character @ may be used to signify all active links. (Partial wildcards, such as PSI@, are *not* allowed, however.)

STATUS Requests status information about the link. May not be used with the TRACE option. For all options, displays the linkname, linktype, and additional information as follows:

ALL Prints the information displayed by the LINKSTATE, CONFIGURATION, and STATISTICS parameters. These parameters are described more fully later in this section. Additional information about the LINKCONTROL command and its parameters can be

found in Appendix A , "LINKCONTROL Command."

`CONFIGURATION` Prints the information displayed by the `LINKSTATE` parameter along with link configuration information for the link. The link configuration data consists mainly of the configuration information that was input for this link during NMMGR configuration. The fields that are displayed by this parameter are described in Appendix A , "LINKCONTROL Command."

`LINKSTATE` Prints link status information, including the link name, link type, and the current status of the link.

`DIAGSTATS` Prints the information displayed by the `LINKSTATE`, `CONFIGURATION`, and `STATISTICS` parameters along with additional diagnostic statistics.

`STATISTICS` Prints the information displayed by the `LINKSTATE` command and link statistics, including accumulated error information. This includes such information as the number of data bytes sent and received, and the number of frames sent and received. The fields that are displayed by this parameter are described in Appendix A , "LINKCONTROL Command."

`RESET` Resets the accumulated data and link statistics that are displayed by the previously described parameters to 0. Displays the same fields that are displayed by the `STATISTICS` parameter.

The `STATUS` and `TRACE` parameters are mutually exclusive and may not be specified together in a `LINKCONTROL` command.

`TRACE` Activates link-level tracing. Only one active trace is allowed per link. If a trace is already active, issuing the command a second time will result in a TRACE REQUEST FAILED error message (NMERR 182).

If `TRACE` is specified, either `ON` or `OFF` must also be specified.

ON	Turns link level tracing on.
OFF	Turns link level tracing off. Any subsequent subparameters are checked for syntax but are otherwise ignored.
DATA	(LAN and token ring links only) Traces all read and write requests. If neither DATA nor ALL is specified, DATA is the default.
ALL	(LAN and token ring links only) Traces all read, write, control, status, and exception requests.
PARTIAL	(Point-to-Point only) Trace all read, write, control, status, and exception requests. Only the first 16 bytes of data are traced for reads or writes.
FULL	(Point-to-Point only) Traces the full data field for all read, write, control, status, and exception requests. If neither PARTIAL nor FULL is specified, PARTIAL is the default.
buffsize	The trace buffer size in memory, in kilobytes. This area is used to buffer trace data before it is written to disk. Allowable values are from 1 to 16.
tracefile	Actual file designator of the disk file where the trace is to be written. If not specified, the trace will automatically be written to a file with the name NMTC*nnnn*.PUB.SYS, where *nnnn* is a value between 0000 and 9999. If the filename is specified without group or account names, the current group and account names are used.

TRACE may not be specified in a LINKCONTROL command also containing STATUS.

Discussion

The LINKCONTROL command returns link statistics and configuration information or activates or deactivates link level tracing on the specified link. The NMCONFIG.PUB.SYS and the link must be active for this command to be operative.

If a trace option that is inapplicable for a certain link is specified for that link, then the default for that link type will be used.

The LINKCONTROL command does not work on an X.25 link because the link is in the DTC. For equivalent functionality, use the OpenView DTC Manager for PC-based X.25 links or TermDSM for host-based X.25 links.

Example 1:

```
LINKCONTROL SYSLINK, ALL

Linkname: SYSLINK   Linktype: IEEE8023   Linkstate: CONNECTED

Physical Path:                  4.3
Inbound Buffer Size:                  1536
Inbound Number of Buffers:              64
Inbound Buffers Available:              56
Current Station Address:              08-00-09-00-EE-8C
Default Station Address:              08-00-09-00-EE-8C
Current Receive Filter:          bad(0) multi(1) broad(1) any(0)
Current Multicast Addresses:
 09-00-09-00-00-01
Transmits no error              19231   Receives no error      2493981
Transmits error                     0   Receives error              16
Out of Tx bufs                      0   Out of Rx bufs               2
Transmits deferred                370   Carrier losses               0
Transmits 1 retry                  16   Reflectometer                0
Transmits 1 retry                  15   CRC errors                  12
Transmits 16 collisions             0   Whole byte errors           11
Transmits late collision            0   Size range errors            0
802 chip restarts                   0   Frame losses                10
Heartbeat losses                    0
```

Example 2:

```
:LINKCONTROL SYSLINK;TRACE=ON,DATA,8
Trace has been successfully enabled for SYSLINK.
The trace file, for SYSLINK, is NMTC0006.PUB.SYS.

:LINKCONTROL SYSLINK;TRACE=OFF
Trace has been successfully disabled for SYSLINK.
```

NETCONTROL

Command used to initialize, terminate, and control the operation of the network transport.

Syntax

```
                {function}[;function]
NETCONTROL      {entity  }[;entity  ]
```

Use

Available	In Session?	YES
	In Job?	YES
	In Break?	NO
	Programmatically?	YES
Breakable?		NO
Capabilities?		NM

Parameters

function Only one of each type of function is recommended on a command line. Refer to function descriptions on the following pages. The functions are:

ADDLINK	TRACEOFF
DELLINK	TRACEON
START	UPDATE
STATUS	VERSION
STOP	

Only one of each type of function (START, TRACE, etc.) is recommended on a command line. For example, the command:

```
:NETCONTROL
TRACEON=HDM;START;TRACEON=HD;NET=LAN1
```

is not recommended because TRACEON appears twice and also appears with START.

entity One or more of the entities defined for NETCONTROL. The keywords for these entities are shown in Figure 7-1.

NET Specifies a group entity that consists of a network interface which is not a gateway half, and all the protocol modules that are configured for that network

interface. Not all functions may be applied as a group; see the individual command functions for details.

`GATE` Specifies a group entity that consists of a configured gateway half network interface, and all the protocol modules that are configured for that network interface. Not all functions may be applied as a group; see the individual command functions for details.

Note: This keyword cannot be used to select true "gateways" as configured in the INTERNET subtree under a network interface.

`PROT` Specifies a particular general protocol module or a particular network interface protocol module upon which a function will act.

`NI` Specifies a particular network interface upon which a function will act. Usually used in conjunction with the `PROT=` keyword to access a particular network interface protocol. See the individual command functions for details.

Discussion

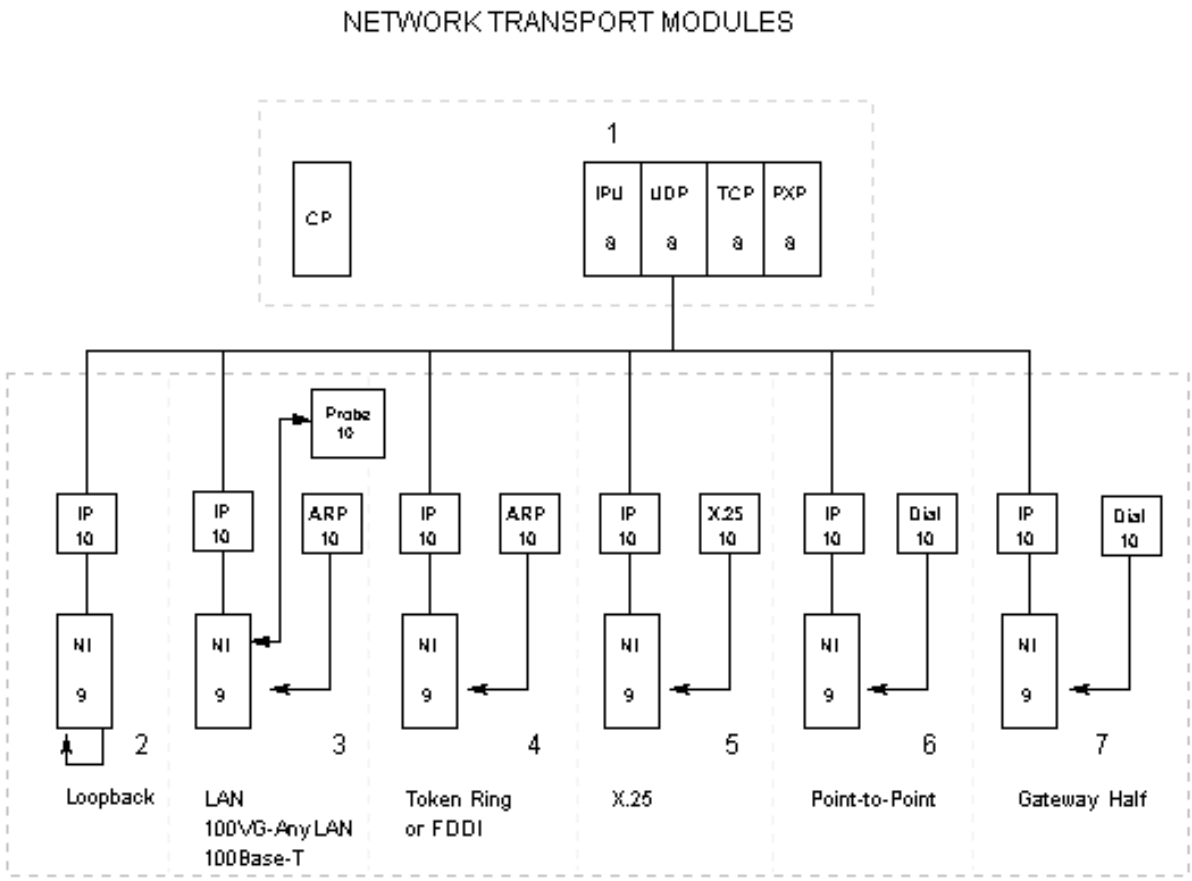
The `NETCONTROL` command is composed of functions (`START`, `STOP`, `STATUS`, etc.) to be executed against one or more entities shown in Figure 7-1.

Notice that the first seven entities are composed of groups of modules. For example, let us look at the third entity, labeled 3. This entity, `NET=`*`niName`* where *`niName`* is the configured LAN *`niName`*, combines the network interface (NI) configured for any LAN link and the protocols configured for that NI, which can include IP, ARP, and Probe.

The remaining three entities, numbers 8–10 in Figure 7-1, allow exact specification of one and only one module of the network transport. This is especially useful when troubleshooting. Refer to `NETCONTROL STATUS` and `NETCONTROL TRACE` for more information.

For information on how the entities are affected by a particular function, refer to the command page for that function.

Figure 7-1 The NETCONTROL Entities



1 General transport: Control Process (CP) and the general protocols (TCP, PXP, UDP, IPU)

2 NET = niName (Loopback -- includes NI and IP

3 NET = niName (LAN -- includes 100VG-AnyLAN, 100Base-T, NI, P, ARP, and PROBE)

4 NET = niName (Token Ring or FDDI -- includes NI, IP, and ARP

5 NET = niName (X.25 -- includes NI, IP, and X.25)

6 NET = niName (Point-to-Point -- includes NI, IP, and Dial)

7 GATE = gateName (Gateway Half -- includes NI, IP and Dial)

8 PROT = gprot (One of TCP, PXP, UDP, or IPU)

9 NI = niName (Loopback, LAN, Token Ring, FDDI, Gateway Half, X.25, or Point-to-Point

10 NI = niName; PROT = niProt (IP only for Loopback;IP, ARP, or PROBE for LAN; IP or ARP for Token Ring or FDDI; IP or Dial for Gateway Half and Point-to-Point; IP or X.25 for X.25)

Example

This example illustrates how each NETCONTROL command is used. See individual commands for further details and examples.

To check the level of transport software installed, enter

```
:NETCONTROL VERSION
```

To start a transport having a LAN network named “LAN1” plus a loopback network named “LOOP” and a router network named “ROUTER1”, having links “PSI40” and “PSI48” under it, enter

```
:NETCONTROL START; NET=LAN1
:NETCONTROL START; NET=ROUTER1
:NETCONTROL START; NET=LOOP
```

To then enable the NS 3000/iX Services (DSCOPY, etc.), enter

```
:NSCONTROL START
```

To now take down the “PSI40” link on the “ROUTER1” network because someone wants to use that link for RJE/iX access, (the other link “PSI48” is still available to the router), enter

```
:NETCONTROL DELLINK=PSI40; NET=ROUTER1
```

To check if the PROBE protocol is running on the “LAN1” network, enter

```
:NETCONTROL STATUS=ALL; NI=LAN1; PROT=PROBE
```

To bring the PSI link “PSI40” back online after RJE/iX users have finished with it, enter

```
:NETCONTROL ADDLINK=PSI40; NET=ROUTER1
```

To update the “ROUTER1” network with new node mappings added to the NMCONFIG file without stopping that network, enter

```
:NETCONTROL UPDATE=MAPPING; NET=ROUTER1
```

To start TCP message and data tracing for all networks (since TCP is a general protocol), enter

```
:NETCONTROL TRACEON=MHD; PROT=TCP
```

To stop the NS 3000/iX Services, enter

```
:NSCONTROL STOP
```

To stop all networks, tracing, and the entire transport, enter

```
:NETCONTROL STOP
```

NETCONTROL ADDLINK

Dynamically adds a configured network link to the active network interface.

Syntax

```
NETCONTROL ADDLINK=linkName; {NET=niName       }
                             {GATE=gatehalfName}
```

Parameters

ADDLINK=*linkName*

Specifies the name of the link to be dynamically added to the specified active NI. The linkname must be a valid NI link name configured in the NMMGR Link Selection screen and also in the Network Interface Links screen under the specified “niname” or “gatehalfname” NI. If already added, an “ALREADY STARTED” error will occur, or if the linkname is not valid, a “NOT CONFIGURED” error will occur.

NET=*niName*

Specifies the name of an active network interface under which the specified linkname is configured. Enter any valid NI name from the NMMGR Network Interface Configuration screen which is not a gateway half. If this NI is not active, a “NOT STARTED” error will occur.

GATE=*gatehalfName*

Specifies the name of an active gateway half network interface under which the specified linkname is configured. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. If this NI is not active, a “NOT STARTED” error will occur.

Discussion

This command adds an already configured link to an active network interface without having to first bring down and then restart the network interface or the entire transport. This can be a link for a newly-configured node mapping, a link being shared with another subsystem such as RJE/iX or SNA/iX, or a link being restarted after an earlier failure due to link errors. Note that some link errors are so serious that an ADDLINK cannot restore use of the link.

The control process will create a new link driver for the specified link and bind it to the existing network interface and its network interface protocols.

This function is mainly used with router NI types.

Example

To add the linkname “PSI48” to the active NI “ROUTER1”, enter

```
:NETCONTROL ADDLINK=PSI48; NET=ROUTER1
```

NETCONTROL DELLINK

Dynamically deletes a configured network link from the active network interface.

Syntax

```
NETCONTROL DELLINK=linkName; {NET =niName      }
                             {GATE=gatehalfName}
```

Parameters

`DELLINK=linkName`

Specifies the name of the link to be dynamically deleted from the specified active NI. The linkname must be a valid NI link name configured in the NMMGR Link Selection screen and also in the Network Interface Links screen under the specified “niname” or “gatehalfname” NI. If already deleted, a “NOT STARTED” error will occur, or if the linkname is not valid, a “NOT CONFIGURED” error will occur.

`NET=niName`

Specifies the name of an active network interface under which the specified linkname is configured. Enter any valid NI name from the NMMGR Network Interface Configuration screen which is not a gateway half. If this NI is not active, a “NOT STARTED” error will occur.

`GATE=gatehalfName`

Specifies the name of an active gateway half network interface under which the specified linkname is configured. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. If this NI is not active, a “NOT STARTED” error will occur.

Discussion

This command deletes a previously configured and started link from an active network interface without having to bring down the entire network interface or transport. This command is particularly useful when making cabling or modem changes, to make a device unusable for security reasons, or when sharing a device with other subsystems such as RJE/iX or SNA/iX. Certain types of errors can also sometimes be cleared by a `DELLINK` followed by an `ADDLINK`.

The control process will unbind the network interface protocols and network interface from the existing link driver, then terminate that link driver. Depending on the link type, the link driver may not actually terminate if other links are still bound. The network interface and its protocols remain active until that NI is stopped using the `NETCONTROL STOP` command.

This function is mainly used with router NI types.

Example

To delete the linkname “PSI48” from the active NI “ROUTER1”, enter

```
:NETCONTROL DELLINK=PSI48; NET=ROUTER1
```

NETCONTROL START

Initiates the network transport, including the control process, general protocols, network interfaces, and their protocols. Also initiates individual network interfaces on an active transport.

Syntax

```
NETCONTROL START  [;{NET=niName       }]
                    {GATE=gatehalfName}
```

Parameters

START — This function, if issued when transport is not active, initializes the control process and general protocols. When NET or GATE is also used, all configured protocols and associated modules for the specified network interface will be initialized as well, however only one such keyword may be specified per command. If you are starting several network interfaces, several commands will be required.

Unless you start network interfaces, no internetwork communications will be possible.

If the general protocols fail to start, a NETCONTROL STOP command may be required before another start can be attempted.

NET=*niName* — Specifies the name of a configured network interface to be started. All protocols and links configured to initially start for that NI will also be started. Enter any valid NI name from the NMMGR Network Interface Configuration screen which is not a gateway half. If neither NET nor GATE are specified, only the control process and general protocols will start. Otherwise if this is the first START, those will be started before the specified NET or GATE. If the specified entity is already running, an "ALREADY STARTED" error will occur.

GATE=*gatehalfName*

Specifies the name of a configured gateway half network interface to be started. All protocols and links configured to initially start for that NI will also be started. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. The *niName* discussion for NET and GATE applies to gatehalf name also.

Discussion

In order for internetwork communications to be possible, you must activate at least one network interface using the NET keyword.

When this command is entered with an X.25 NI name, the system accesses the DTC/X.25 Network Access subsystem to verify that the X.25 line is started. If the line is not started, the command fails. If the X.25 line is started, the command is successfully completed if everything is correct. This enables the X.25 address that is associated with this system in the DTC/X.25 Network Access, and connections can be generated or accepted from this system.

Example 1

In Example 1, the node has one LAN link configured (LAN1) plus loopback (LOOP). Starting the network requires issuing a NETCONTROL START for each configured network interface (NET=*niName*). Once both network interfaces (and related entities) of the network transport have been successfully initiated, as indicated by the lack of error messages, any other related subsystems installed on the node can be initiated. This node, as is typically the case, has NS 3000/iX Services installed.

```
:NETCONTROL START;NET=LAN1
:NETCONTROL START;NET=LOOP
:NSCONTROL START
```

Refer to the NSCONTROL command pages in this section for more information.

Be aware that to successfully initialize a node, the commands must be issued in the order shown: first all required NETCONTROL commands, then any NSCONTROL commands.

This first example provides an overview of initializing a node, showing where NETCONTROL fits into the process. The next five examples examine the START function and how it affects the entities defined for initialization (Figure 7-1). As will be shown in the examples, the keywords included with the START function and the entities affected determine which events occur at initialization. To understand this relationship, it is helpful to see the events that occur when the network transport is initialized.

Example 2

Example 2 shows the events associated with the START function at initiation. As indicated in the status report, the general transport is not active. Therefore, the first events of initiation are to initialize the control process (CP) and the general protocols. Compare the displayed events to the defined entities of Figure 7-1. The events displayed in this example show creation of the general protocols. The START function always creates the control process and the general protocols, if they do not already exist, before acting on any of the other entities.

```
:NETCONTROL STATUS
TRANSPORT NOT ACTIVE.  (NETXPORTWARN 0001)
ENCOUNTERED ONE OR MORE WARNINGS WHILE PROCESSING COMMAND.  (CIWARN 4437)

:NETCONTROL START
** NETXPORT Control Process; Transport start
- Loc: 50; Class: 4; Parm= $0000002C; PIN: 44
** NETXPORT TCP; General protocol start
- Loc: 18501; Class: 4; Parm= $00865910; PortID: $FFFFDFF1
** NETXPORT UDP; General protocol start
- Loc: 19; Class: 4; Parm= $00000000; PortID: $FFFFDFF2
** NETXPORT IP Update; General protocol start
- Loc: 3; Class: 4; Parm= $00000000; PortID: $FFFFDFF4
** NETXPORT Net Timers; Starting
- Loc: 4440; Class: 4; Parm= $00000000; PortID: $FFFFDFED
```

The initiation events shown in this example are always executed for the first NETCONTROL START command, whether or not a network interface is specified. However, once the general transport is initialized, subsequent NETCONTROL START commands do not change the modules of the general transport.

Example 3

Example 3 displays the error message that will appear if you issue a NETCONTROL START command when the control process is already initialized.

```
:NETCONTROL START
ALREADY STARTED.  (NETXPORTERR 4045)
ENCOUNTERED ONE OR MORE ERRORS WHILE PROCESSING COMMAND.  (CIERR 4436)
```

Example 4

In Example 4, the LAN NI, configured as LAN1, is started on the first NETCONTROL START command. Notice that the initiation events to initialize the general protocols are immediately followed by the start of the LAN NI with its associated protocols: IP, probe, and ARP. Compare the displayed events to the defined entities of Figure 7-1. The events displayed show creation of the control process, the general protocols, and the LAN NI entities.

```
:NETCONTROL START;NET=LAN1
** NETXPORT Control Process; Transport start
- Loc: 50; Class: 4; Parm= $0000002C; PIN: 44
** NETXPORT TCP; General protocol start
- Loc: 18501; Class: 4; Parm= $00865910; PortID: $FFFFDFF1
** NETXPORT UDP; General protocol start
- Loc: 19; Class: 4; Parm= $00000000; PortID: $FFFFDFF2
** NETXPORT IP Update; General protocol start
- Loc: 3; Class: 4; Parm= $00000000; PortID: $FFFFDFF4
** NETXPORT Net Timers; Starting
- Loc: 4440; Class: 4; Parm= $00000000; PortID: $FFFFDFED
** NETXPORT Map Tbl; Mapping Table Created
- Loc: 1; Class: 4; Parm= $95C80250; PortID: $95C80250
** NETXPORT LAN NI; Network interface start
- Loc: 28; Class: 4; Parm= $95CC8000; PortID: $FFFFFE88
** NETXPORT IP; Protocol start
- Loc: 102; Class: 4; Parm= $D4FD8000; PortID: $FFFFFE84
** NETXPORT Probe; Protocol start
- Loc: 35; Class: 4; Parm= $00000000; PortID: $FFFFDFF3
** NETXPORT ARP; Protocol start
- Loc: 3; Class: 4; Parm= $00000000; PortID: $FFFFDFF5
```

Example 5

Example 5 shows the initiation events for the loopback network interface. For this example, the loopback NI is configured as LOOP and the general protocols are already active.

```
:NETCONTROL START;NET=LOOP
** NETXPORT Map Tbl; Mapping Table Created
- Loc: 1; Class: 4; Parm= $D5208250; Pin: 0
** NETXPORT Loopback NI; Network interface start
- Loc: 28; Class: 4; Parm= $96038000; PortID: $FFFFFE8A
** NETXPORT IP; Protocol start
- Loc: 102; Class: 4; Parm= $D5218000; PortID: $FFFFFE89
```

Notice that only the Loopback NI and its associated protocol, Internet Protocol (IP), are started; there was a previously issued NETCONTROL START command. Compare the displayed events to the defined entities of Figure 7-1. The events displayed show creation of the Loopback NI entity.

Example 6

Starting the LAN NI, configured as LAN1, when the general protocols are already active, gives you the following:

```
:NETCONTROL START;NET=LAN1
** NETXPORT Map Tbl; Mapping Table Created
- Loc: 1; Class: 4; Parm= $D5C80250; Pin: 0
** NETXPORT LAN NI; Network interface start
- Loc: 28; Class: 4; Parm= $96430000; PortID: $FFFFFE81
** NETXPORT IP; Protocol start
- Loc: 102; Class: 4; Parm= $D5CD0000; PortID: $FFFFFE88
** NETXPORT Probe; Protocol start
- Loc: 35; Class: 4; Parm= $00000000; PortID: $FFFFDFF3
** NETXPORT ARP; Protocol start
- Loc: 3; Class: 4; Parm= $00000000; PortID: $FFFFDFF5
```

Notice that only the LAN NI and its associated protocols are started. Compare the displayed events to the defined entities of Figure 7-1. The events displayed show creation of the LAN NI entity.

NETCONTROL STATUS

Displays status and configuration information for the transport entity specified.

Syntax

```
NETCONTROL STATUS[=ALL] [;{NI=niName [;PROT=niProt]} ]
                        {NET=niname              }
                        {GATE=gatehalfname       }
                        {PROT=gProt              }
```

Parameters

`STATUS[=ALL]` Specifies that any additional status information should be displayed, if additional data is available beyond the default.

`NI=`*`niname`*

Specifies the name of a configured network interface to display the status of. Enter any valid NI name from the NMMGR Network Interface screen which is not a gateway half. If the specified NI was not previously configured and started, an “ENTITY NOT ACTIVE” error will occur. If transport was not previously started, a “TRANSPORT NOT ACTIVE” warning will occur.

Specifying `NI=niname` without the `;PROT=` option displays status for the network interface itself.

`NET=`*`niName`*

Specifies the name of a configured network interface which is not a gatehalf. Enter any valid NI name, as configured with NMMGR.

`GATE=`*`gatehalfname`*

Specifies the name of a configured gateway half network interface to display the status of. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. If the specified gatehalf NI was not previously configured and started, an “ENTITY NOT ACTIVE” error will occur. If transport was not previously started, a “TRANSPORT NOT ACTIVE” warning will occur.

PROT=*gProt* PROT=*niProt*

Specifies that a protocol is the pertinent entity for each specified function to act on. Enter the name of the protocol, as follows:

gprot Specifies the name of one transport general protocol to display the status of. Valid inputs are TCP, UDP, PXP, or IPU. If the specified protocol did not start or is not one of these inputs, an "ENTITY NOT ACTIVE" error will occur. If transport was not previously started, a "TRANSPORT NOT ACTIVE" warning will occur.

niprot Specifies the name of one network interface protocol to display the status of; must be used in conjunction with the NI=niname parameter. Valid inputs depend on the NI type, as shown here. If the specified protocol did not start, is not configured, or is not one of these inputs, an "ENTITY NOT ACTIVE" error will occur. If transport was not previously started, a "TRANSPORT NOT ACTIVE" warning will occur.

NI Type:	**Valid Network Interface Protocol Names**
LAN	IP, PROBE, ARP
TOKEN	IP, ARP
FDDI	IP, ARP
100VG-AnyLAN	IP, PROBE, ARP
100Base-T	IP, PROBE, ARP
ROUTER	IP, DIAL
X.25	IP, X25
GATEHALF	IP, DIAL
LOOP	IP

Discussion

This command displays status and configuration data for the active transport, using several different output formats depending on the keywords specified. Some of the formats are specific to the control process, a network interface, or a specific protocol. Any entities which are not active cannot have their status displayed.

This command differs from other NETCONTROL commands in that it produces warnings, not errors, if transport is not active. This is often used to determine if transport as a whole is running or not.

NOTE HP does not recommend combining this function on the same command line as other functions, in an attempt to determine if the other function worked.

NOTE The output format of all NETCONTROL commands is subject to change without notice. Programs which are developed to postprocess NETCONTROL output should not depend on the exact format (spacing, alignment, number of lines, upper or lower case, or spelling) of any NETCONTROL command output.

Example 1

Example 1 is a sample of the output that is displayed when the NETCONTROL STATUS command is issued without specifying a network interface or general protocol.

```
:NETCONTROL STATUS
GENERAL TRANSPORT STATUS    : MON, FEB 17, 1992,  8:49 AM
TRANSPORT STARTED           : MON, FEB  3, 1992,  2:25 PM

FLAGS                       : $000014C0
MAX NETWORK INTERFACES      : 32
MAX NODE NAMES              : 360
LOG ID                      : $00040003
TRACE ID                    : $00000000
CONTROL PROCESS PORT ID     : $FFFFFF37

HOME NETWORK                : LAN1
CONFIGURATION FILE          : NMCONFIG.PUB.SYS
TRACE MASK                  : $00000000
NODE NAME                   : NODEA.XLNET.ACCTG
```

Example 2

Example 2 is a sample of the output that is displayed when you issue the NETCONTROL STATUS command specifying the LAN1 network interface via the NI= parameter.

```
:NETCONTROL STATUS;NI=LAN1

NETWORK INTERFACE REPORT       : MON, FEB 17, 1992,  8:52 AM
NETWORK INTERFACE STARTED      : MON, FEB  3, 1992,  2:33 PM
FLAGS                          : $00000006
NIB - PCB LINK INFO            : FIRST $452C4290
NIB - NIB LINK INFO            : NEXT $00000000
NI PROTOCOLS                   : CURRENT $00000000  MAXIMUM $00000004
MAPPING TABLE SIZE             : $00000400
MAPPING TABLE ID               : $C8B80250
OUTBOUND BUFFERS               : SIZE $000005EA  NUMBER $00000100
NETWORK INTERFACE TYPE         : LAN
PORT ID                        : $FFFFFF21
WRITE BUFFER INFO              : POOL $0000000A
STORE/FORWARD BUFFER INFO      : POOL $0000000A
TRACE ID'S                     : TRACE $00000000
NAME                           : LAN1
NETWORK IP ADDRESS             : $0F0D7033   15.13.112.51
NETWORK SUBNET MASK            : $FFFFF800   255.255.248.0
TRACE MASK                     : $00000000

DEVICE INFORMATION             :

DEVICE                         : SYSLINK (# 0)
DEVICE TYPE                    :
LINK BUFFER SIZE               : $000005EA
PROTOCOLS CONNECTED            : $00000004
PHONE NUMBER INDEX             : $00000000
TRAN PORT INFO                 : PORT ID $FFFFFF20
```

Example 3

Example 3 is a sample of the output that is displayed when the NETCONTROL STATUS command is issued and the LAN1 network interface and the PROBE protocol are specified.

```
:NETCONTROL STATUS;NI=LAN1;PROT=PROBE
NETWORK INTERFACE PROTOCOL STATUS : WED, JAN 19, 1994,  3:31 PM
PROTOCOL STARTED   : WED, JAN 19, 1994   4:23 PM

PROTOCOL NAME          : PROBE
PROTOCOL ID            : $00000503
PROTOCOL FLAGS         : $00000000
TRACE MASK             : $00000000

PCB LINK INFO          : NEXT $4533FA58
TRACE ID'S             : TRACE $00000000
PORT ID                : $FFFFC77
NETWORK NAME           : LAN1
```

Example 4

Example 4 is a sample of the output that is displayed when the NETCONTROL STATUS command is used specifying the LAN1 network interface via the NET= parameter.

```
:NETCONTROL STATUS;NET=LAN1
NETWORK STATUS : WED, JAN 19, 1994,  3:31 PM
PROTOCOL STARTED : WED, JAN 19, 1994,  4:23 AM

PROTOCOL NAME    : IP
PROTOCOL ID      : $00000500
PROTOCOL FLAGS   : $00000000
TRACE MASK       : $00000000

PCB LINK INFO    : NEXT $00000000
TRACE ID'S       : TRACE $00000000
PORT ID          : $FFFFFC78
NETWORK NAME     : LAN1
NETOWRK IP ADDRESS        : $0C0E84A6   12.14.132.166
NETWORK SUBNET MASK       : $FF000000   255.0.0.0
```

Example 5

Example 5 is a sample of the output that is displayed when the NETCONTROL STATUS command is issued and the LAN1 network interface and the ARP protocol is specified.

```
:NETCONTROL STATUS; NI=LAN1; PROT=ARP

NETWORK INTERFACE PROTOCOL STATUS : WED, JAN 26,
1994,  1:55 PM
PROTOCOL STARTED : WED, JAN 26, 1994,  1:55 PM

PROTOCOL NAME    : ARP
PROTOCOL ID      : $00000508
PROTOCOL FLAGS   : $00000000
TRACE MASK       : $00000000

PCB LINK INFO    : NEXT $4533FDC0
TRACE ID'S       : TRACE $00000000
PORT ID          : $FFFFFF10
NETWORK NAME     : LAN1
```

To report the status of the control process, enter

```
:NETCONTROL STATUS

GENERAL TRANSPORT STATUS : WED, JAN 26, 1994,  9:12 AM
TRANSPORT STARTED        : WED, JAN 26, 1994,  3:57 AM

FLAGS                    : $000014C0
MAX NETWORK INTERFACES   : 32
MAX NODE NAMES           : 360
LOG ID                   : $00000003
TRACE ID                 : $00010081
CONTROL PROCESS PORT ID  : $FFFFFC8E

HOME NETWORK             : LAN1
CONFIGURATION FILE       : NMCONFIG.PUB.SYS
TRACE MASK               : $00000040
NODE NAME                : NODE.DOMAIN.ORG
```

If the control process is not active, you will see a warning

```
:NETCONTROL STOP
:NETCONTROL STATUS
TRANSPORT NOT ACTIVE. (NETXPORTWARN 0001)
```

NOTE The output format of all NETCONTROL commands is subject to change without notice. Programs which are developed to postprocess NETCONTROL output should not depend on the exact format (spacing, alignment, number of lines, upper or lower case, or spelling) of any NETCONTROL command output.

NETCONTROL STOP

Terminates individual network interfaces on an active transport, or the entire transport and all its network interfaces.

Syntax

```
NETCONTROL STOP [;{NET=niName       }]
                 {GATE=gatehalfName}
```

Parameters

STOP — This function, if issued without parameters when transport is active, irrecoverably stops the entire transport. When NET or GATE is specified, only that one network interface and its attached protocols are terminated; all other network interfaces and protocols will continue to operate.

NET=*niName*

Specifies the name of a configured network interface to be terminated, which was previously started. All protocols and links configured under that NI will also be terminated. Enter any valid NI name from the NMMGR Network Interface screen which is not a gateway half. If neither NET nor GATE are specified, the entire transport will be terminated, including all links, NIs, protocols, and the control process. If transport or the specified NI was not running, a “NOT STARTED” error will occur.

GATE=*gatehalfName*

Specifies the name of a configured gateway half network interface to be terminated, and which was previously started. All protocols and links configured under that NI will also be terminated. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. The *niName* discussion for NET and GATE applies to gatehalf name also.

Discussion

If STOP is issued without the NET or GATE keyword, all entities of the network transport are terminated. If STOP is combined with the keyword NET, only the specified non-gatehalf network interface is terminated. If STOP is combined with the keyword GATE, only the specified gateway half is terminated.

When this command is entered with an X.25 NI name, the system accesses the DTC/X.25 Network Access Subsystem to disable the X.25 address that is associated with this system. The DTC/X.25 Network Access then no longer accepts connections for this system. There is no impact on the connections established between any other systems and the DTC/X.25 Network Access.

Example 1

Example 1 shows how NETCONTROL fits into the process of terminating a node. In example 1, the node has an NS 3000/iX Link. The NSCONTROL command prevents users or programs from accessing any network services. (Refer to the NSCONTROL command page in this section for a complete description of NSCONTROL.) NETCONTROL STOP terminates all active entities of the network transport.

NOTE When multiple NETCONTROL or NSCONTROL commands are embedded in UDC's, commandfiles, or batch jobs, HP recommends the use of :PAUSE commands between commands, to give each time to execute before starting the next command.

```
:NSCONTROL STOP
:NSCONTROL ABORT
:NETCONTROL STOP
```

Example 2

Example 2 shows which network transport entities are affected by the STOP function. As will be shown in examples 3 and 4, the keywords included with the STOP function determine which entities are affected.

```
:NETCONTROL STOP
** NETXPORT ARP; Protocol stop
- Loc: 27; Class: 4; Parm= $00000000; PortID: $FFFFFE89
** NETXPORT Probe; Protocol stop
- Loc: 37; Class: 4; Parm= $00000000; PortID: $FFFFFE89
** NETXPORT IP; Protocol stop
- Loc: 105; Class: 4; Parm= $00000000; PortID: $FFFFFE89
** NETXPORT Control Process; Device Shutdown Warning
- Loc: 283; Class: 3; Parm= $04F502E6; PortID: $FFFFFE8A
** NETXPORT LAN NI; Network interface stop
- Loc: 29; Class: 4; Parm= $96430000; PortID: $FFFFFE81
** NETXPORT Map Tbl; Mapping Table Deleted
```

```
- Loc: 2; Class: 4; Parm= $D5208250; PortID: $D5208250
** NETXPORT IP Update; General protocol stop
- Loc: 19; Class: 4; Parm= $00000000; PortID: $FFFFFE88
** NETXPORT UDP; General protocol stop
- Loc: 25; Class: 4; Parm= $00000000; PortID: $FFFFDFF3
** NETXPORT Net Timers; Stopping
- Loc: 4040; Class: 4; Parm= $96430000; PortID: $FFFFFE81
** NETXPORT Control Process; Transport stop
- Loc: 51; Class: 4; Parm= $00000000; PortID: $FFFFDFF0
```

Example 3

Example 3 shows what happens if the general transport and both network interfaces are active, and the user specifies the Loopback NI. Notice that the STOP function acts only on the Loopback NI entity. The general transport is still active.

```
:NETCONTROL STOP;NET=LOOP
** NETXPORT IP; Protocol stop
- Loc: 105; Class: 4; Parm= $00000000; PortID: $FFFFFE7F
** NETXPORT Loopback NI; Network interface stop
- Loc: 29; Class: 4; Parm= $96F80000; PortID: $FFFFFE88
** NETXPORT Map Tbl; Mapping Table Deleted
- Loc: 2; Class: 4; Parm= $96F78250; Pin: 0
```

Example 4

In Example 4, only the general transport and the LAN are active. The STOP function terminates the LAN NI entity. The general transport is still active.

```
:NETCONTROL STOP;NET=LAN1
** NETXPORT ARP; Protocol stop
- Loc: 27; Class: 4; Parm= $00000000; PortID: $FFFFFF04
** NETXPORT Probe; Protocol stop
- Loc: 37; Class: 4; Parm= $00000000; PortID: $FFFFDFF0
** NETXPORT IP; Protocol stop
- Loc: 105; Class: 4; Parm= $00000000; PortID: $FFFFFE84
** NETXPORT Control Process; Device Shutdown Warning
- Loc: 283; Class: 3; Parm= $04F502E6; PortID: $FFFFFE8A
** NETXPORT LAN NI; Network interface stop
- Loc: 29; Class: 4; Parm= $D8020000; PortID: $FFFFFE85
** NETXPORT Map Tbl; Mapping Table Deleted
- Loc: 2; Class: 4; Parm= $97480250; PortID: $97480250
```

NETCONTROL TRACEON and TRACEOFF

Enables or disables message tracing for the specified transport entity.

Syntax

```
NETCONTROL {TRACEON=type[,options]} [;{NI=niName [;PROT=niprot]} ]
           {TRACEOFF              }   {NET=niName              }
                                      {GATE=gatehalfname         }
                                      {PROT=gprot                }
```

where the parameter option has the following options:

```
    [DISC][,[filename][,[recsize][,filesize] ]]
```

Parameters

`TRACEON` Enables tracing for the one entity specified by the NI, `PROT`, `NET`, or `GATE` keywords, or for the control process if none of those keywords are specified. The control process will be started if it is not already running. This function cannot be used to modify any parameters of tracing which has already been enabled. If tracing is already enabled for the specified entity, a "PREVIOUSLY ENABLED" error will occur.

`type` (Required). Specifies the type of data to trace from the specified entity. This field is made up of one or more of the following key letters, concatenated, and entered in any order:

`M` — Trace Messages
`H` — Trace Packet Header Data
`D` — Trace Packet Data
`S` — Trace State Transitions
`B` — Trace Buffers
`N` — Trace Nodal Management Events

Recommended type setting is MHD. There is no default.

`options` Specifies additional information about where to put the collected trace data. There are several parameters.

NOTE A comma *must* precede a parameter whenever (a) that parameter is included or (b) that parameter is omitted but any *other* parameter which follows it is included.

`DISC` (Optional). Trace information will be written to a disc file, specified by the

filename parameter. DISC is the default and the only valid input.

NOTE Tracing to tape is no longer available on MPE/iX.

filename (Optional). The name of the file to which trace data will be written. The default is to automatically create the next highest numbered NMTC*nnnn*.PUB.SYS file, where *nnnn* is a 4-digit number, for each TRACEON command entered.

If you wish several TRACEON commands to trace to the same file, you must specify that filename using this parameter. You may choose an automatically created file for this purpose.

recsize (Optional). Logical record size of the records in the file to which trace data will be written, in number of 16-bit words. This is an internal limit for the tracing facility; the physical record size is always 128. Valid range is 5<=recsize<=1024. Default is 128.

filesize (Optional). Maximum number of records in the trace file. When this limit is reached, the file “wraps”, and tracing continues. The valid range is 32<=filesize<=32000. Default is 1024.

TRACEOFF Disables previously enabled tracing for one entity, which is specified by the NI, PROT, NET, or GATE keywords, or for the control process if none of those keywords are specified. If tracing is not enabled for the specified entity, a “NOT TRACING” error will occur.

NI=*niname*

Specifies the name of a configured network interface the trace will apply to. Enter any valid NI name from the NMMGR Network Interface screen which is not a gateway half. If the specified NI was not previously configured and started, a “NOT STARTED” error will occur.

Specifying NI=niname without the ;PROT= option, or NET=niname, starts tracing for the network interface itself.

NET=*niName*

Specifies the name of a configured network interface which is not a gatehalf. Enter any valid NI name, as configured with NMMGR. Using this parameter, the function applies only to the network interface itself, not to any attached protocols.

GATE=*gatehalfName*

Specifies the name of a configured gateway half network interface to start tracing on. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. If the specified gatehalf NI was not previously configured and started, a "NOT STARTED" error will occur.

PROT=*gprot* PROT=*niProt*

Specifies that a protocol is the pertinent entity for each specified function to act on. Enter the name of the protocol, as follows:

gprot	Specifies the name of one transport general protocol to start tracing on. Valid inputs are TCP, UDP, PXP, or IPU. If the specified protocol did not start or is not one of these inputs, a "NOT ACTIVE" error will occur.
niprot	Specifies the name of one network interface protocol to start tracing on; must be used in conjunction with the NI=niname parameter. Valid inputs depend on the NI type, as shown here. If the specified protocol did not start, is not configured, or is not one of these inputs, a "NOT ACTIVE" error will occur.

NI Type:	Valid Network Interface Protocol Names
LAN	IP, PROBE, ARP
TOKEN	IP, ARP
FDDI	IP, ARP
100VG-AnyLAN	IP, PROBE, ARP
100Base-T	IP, PROBE, ARP
ROUTER	IP, DIAL
X.25	IP, X25
GATEHALF	IP, DIAL
LOOP	IP

Discussion

The tracing functions allow you to enable collection of internal information about what the various transport modules are doing, or what packets are being sent and received at the transport level.

Using TRACEON you instruct a specific module not only to begin tracing, but also what kind of data to trace and what file to put it in. Tracing continues until explicitly stopped via a matching TRACEOFF command, or until the specified module, or all of transport, is stopped. If multiple modules had tracing enabled to capture a problem, stopping transport is the usual way to stop all tracing.

For most problems you will need to enable TCP tracing, and for IP store-and-forward problems you should enable IP tracing; see the examples for sample commands. For link-related problems you should enable link tracing (see the LINKCONTROL command). Other NS tracing can be enabled under the guidance of your HP support representative.

When tracing is enabled successfully, the name of the active trace file is displayed. You should write this down as it will not be repeated at TRACEOFF time; otherwise, to determine which trace file contains the desired data, check trace file creation times by using :LISTF NMTC####.PUB.SYS,3.

As soon as your problem has been duplicated, you should stop tracing to avoid having the file “wrap” and overwrite the data. At completion of tracing, a trace file may be formatted using the NMDUMP.PUB.SYS utility. Much of the information traced will be meaningful only to HP support personnel.

Example

To enable TCP tracing, enter

```
:NETCONTROL TRACEON=MHD; PROT=TCP
TRACE FILE IS NMTC0128.PUB.SYS. (NETXPORT 2000)
```

To disable TCP tracing, enter

```
:NETCONTROL TRACEOFF; PROT=TCP
```

To enable control process tracing, TCP tracing, and IP tracing on the "LAN1" NI, all to the same file, enter

```
:NETCONTROL START; NET=LAN1
:NETCONTROL TRACEON=MHDSBN
TRACE FILE IS NMTC0129.PUB.SYS. (NETXPORT 2000)
:NETCONTROL TRACEON=MHD,DISC,NMTC0129.PUB.SYS; PROT=TCP
TRACE FILE IS NMTC0129.PUB.SYS. (NETXPORT 2000)
:NETCONTROL TRACEON=MHD,DISC,NMTC0129.PUB.SYS; NI=LAN1; PROT=IP
TRACE FILE IS NMTC0129.PUB.SYS. (NETXPORT 2000)
```

To disable all this tracing once enabled, enter

```
:NETCONTROL TRACEOFF; NI=LAN1; PROT=IP
:NETCONTROL TRACEOFF; PROT=TCP
:NETCONTROL TRACEOFF
```

NETCONTROL UPDATE

Dynamically updates selected network transport parameters and configuration information.

Syntax

```
                    {INTERNET}
                    {MAPPING }
NETCONTROL UPDATE= {NETDIR  }; {NET=niName       }
                    {X25     }  {GATE=gatehalfName}
                    {ALL     }
```

Parameters

```
          {INTERNET}
          {MAPPING } Specifies which configuration areas will be dynamically
UPDATE = {NETDIR  } updated. The areas possible depends on the network type.
          {X25     }
          {ALL     }
```

INTERNET — Adds to IPU all gateway data currently configured for the specified network interface or gateway half, meaning all gateways appearing in the NMMGR Neighbor Gateways screen (subtree NETXPORT.NI.niname.INTERNET) and all Reachable Network data configured under each of those gateways. These screens contain information describing the gateways for all directly connected networks and gateway halves, as well as all networks the gateways can reach. Valid for all NI types except Loopback.

MAPPING — Adds all router mappings currently configured for the specified router network interface, to that NI's mapping table, meaning all mappings appearing in that NI's NMMGR Point-to-Point Mapping Configuration screen and the Point-to-Point Reachable Nodes screens under it. Information will be overlaid based on matching IP-Device mapping records. This allows changing routes as well as adding new reachable nodes. Valid for router NI types only.

NETDIR — Adds all currently configured Network Directory (NSDIR.NET.SYS) entries, whose address types apply to the specified network interface's type, to the mapping table for that NI. Valid for LAN, FDDI, 100VG-AnyLAN, 100Base-T and Token Ring NI types only.

X25 — Adds all currently configured Network Directory (NSDIR.NET.SYS) entries having X.25 or IP address types and matching entries in the NMMGR X.25 SVC Address Key Paths screen, to the specified X.25 network interface's mapping table and X.25 protocol module. This allows adding new SVC destinations or adding a new node to the L.U.G. (Local User Group) table. Valid for X.25 NI types only.

ALL — The control process will update all areas which apply to the specified network interface or gateway half's type. Areas not supported for that NI type will not be updated. Updating will occur in this order: INTERNET, MAPPING, NETDIR, X25.

NET=*niName*

Specifies the name of a configured network interface to be updated, which has already been started. Enter any valid NI name from the NMMGR Network Interface Configuration screen which is not a gateway half. If the specified NI is not configured and started, a "NOT STARTED" error will occur.

GATE=*gatehalfName*

Specifies the name of a configured gateway half network interface to be updated, which has already been started. Enter any valid gatehalf NI name from the NMMGR Network Interface Configuration screen. If the specified NI is not configured and started, a "NOT STARTED" error will occur.

Discussion

The update function updates transport with certain configuration changes already made through NMMGR. In this way, those kinds of changes can become active without having to first take down and then restart the network or the entire transport.

The types of changes which can be updated are those concerned with addresses of reachable nodes and networks ONLY; others, such as timeout changes, require stopping and restarting transport to take effect. UPDATE's keywords (NETDIR, INTERNET, etc.) localize which kind of configuration data will be updated. This command can be entered at any time after the specified NI has been started.

Not all options are valid for all network interface types. Table 7-2 summarizes the applicability of the various UPDATE options to each NI type.

Table 7-2 NETCONTROL Update

NI Type	Valid Update Options
LAN	INTERNET, NETDIR, ALL
TOKEN	INTERNET, NETDIR, ALL
FDDI	INTERNET, NETDIR, ALL
100VG-AnyLAN	INTERNET, NETDIR, ALL
100Base-T	INTERNET, NETDIR, ALL
ROUTER	INTERNET, MAPPING, ALL
X.25	INTERNET, X25, ALL
GATEHALF	INTERNET, ALL
LOOP	ALL

NOTE Dynamic updating is additive, so obsolete data can accumulate, possibly resulting in table overflows. If table overflows do occur which prevent access to the desired nodes, transport must be stopped and restarted to clear the condition.

Example

To update the “LAN1” network with new node addresses just added to the Network Directory (NSDIR.NET.SYS), enter

```
:NETCONTROL UPDATE=NETDIR; NET=LAN1
```

NETCONTROL VERSION

Displays the version numbers for the network transport software modules.

Syntax

```
NETCONTROL VERSION[=MOD]
```

Parameters

VERSION[=MOD]	Displays the overall version of the network transport. If qualified with the MOD keyword, displays the version of each of the software modules of the network transport and the overall version.

Discussion

The VERSION function of the NETCONTROL command allows you to check the version numbers of the network transport modules to ensure that they are compatible and up-to-date, or simply to confirm which version is installed on your system. Unlike most other NETCONTROL commands, transport does not need to be started to use this command.

Output from this command is the same as that produced by the NMMAINT.PUB.SYS utility.

Example 1

Example 1 shows how to display the overall version number of the network transport.

```
:NETCONTROL VERSION
NS3000/iX Transport 32098-20033  overall version = B.05.07
```

Example 2

To look at the version numbers of the individual modules, you specify the MOD keyword. You will see a display like the one shown in example 2. Note that the version numbers shown here are only examples, and should not be used to check any actual installation.

```
:NETCONTROL VERSION=MOD

NS3000/iX Transport 32098-20033  module versions:

NM program file:    NETCP.NET.SYS        Version:  B0507048
NL procedure:       NET_CF_VERS          Version:  B0700013
NL procedure:       NET_IPC_VERS         Version:  B0507029
NL procedure:       NET_IPC_VERS2        Version:  B0507012
NL procedure:       NET_IPC_VERS3        Version:  B0507012
NL procedure:       NET_IPC_VERS4        Version:  B0507011
NL procedure:       SIVERS               Version:  B0507009
NL procedure:       PIVERS               Version:  B0507012
Catalog file:       SOCKCAT.NET.SYS      Version:  B0507001
CM program file:    SOCKREG.NET.SYS      Version:  B0507003
NL procedure:       NWTMVERS             Version:  B0507007
NL procedure:       TI_T1_VERS           Version:  B0507001
NM program file:    PT2PNSTN.NET.SYS     Version:  B0507001
Catalog file:       NETMSG.NET.SYS       Version:  B0507022
SL procedure:       NET'UI'VERS          Version:  B0507014
SL procedure:       NET'SL'VERS          Version:  B0507009
NL procedure:       NET_NI_VERS          Version:  B0507016
SL procedure:       NET'PROBE'VERS       Version:  B0507001
NL procedure:       NET_ARP_VERS         Version:  B0507010
SL procedure:       NET'DIAL'VERS        Version:  B0507010
NM program file:    TCPSIP.NET.SYS       Version:  B0507000
SL procedure:       NET'STUB'VERS        Version:  B0507016
NL procedure:       NET_TCP_VERS         Version:  B0507131
NL procedure:       NET_UDP_VERS         Version:  B0507013
NL procedure:       NET_DICT_VERS        Version:  B0507000
SL procedure:       NET'PXP0'VERS        Version:  B0507002
SL procedure:       NET'PXP1'VERS        Version:  B0507004
NL procedure:       NET_IP_VERS          Version:  B0507019
SL procedure:       NET'IPU'VERS         Version:  B0507006
NL procedure:       NET_X25_VERS         Version:  B0507016
SL procedure:       NET'PD'VERS          Version:  B0507024
NL procedure:       NET_PD_VERS          Version:  B0507030
NL procedure:       NET_MAP_VERS         Version:  B0507057
NL procedure:       NET_GLBL_VERS        Version:  B0507024
NL procedure:       NET_REG_VERS         Version:  B0507000
SL procedure:       NET'REG'CM'VERS      Version:  B0507000
SL procedure:       DCLDM_FMT_VERS       Version:  B0507000
NL procedure:       DCLDM_PS_VERS        Version:  B0507000
NL procedure:       DCLDM_CONF_VERS      Version:  B0507000
NL procedure:       NSLOPENLINK_VERS     Version:  B0507003
NL procedure:       RLM_SERVER_VERS      Version:  B0507003
NL procedure:       RLM_CONFIG_VERS      Version:  B0507002
```

```
NL procedure:       RLM_LOAD_TABLE_VERS  Version:  B0507000
SL procedure:       RLM_FMT_VERS         Version:  B0507002
NL procedure:       NET_FC_VERS          Version:  B0507003
SL procedure:       SOCKIOVERS           Version:  B0507016
SL procedure:       SOCKACCESSVERS       Version:  B0507016
SL procedure:       SOCKMISC1VERS        Version:  B0507015
SL procedure:       SUBSYS3FMTVERS       Version:  B0507004
SL procedure:       SUBSYS5FMTVERS       Version:  B0507001
NL procedure:       LEVEL2_RESOLVE_VERS  Version:  B0507002
NM program file:    ICMPSERV.NET.SYS     Version:  B0507004
NM program file:    NETTOOL.NET.SYS      Version:  B0507011
NL procedure:       NETTMRVERS           Version:  B0507036

NS 3000/iX Transport 32098-20033  overall version = B.05.07
```

NSCONTROL

Initiates, terminates, and controls the Network Services subsystem of NS 3000/iX.

Syntax

NSCONTROL *function*[;*function*]…

Use

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capabilities?		NM

Parameters

function Only one of each type of function is recommended on a command line. Refer to function descriptions on the following pages. The functions are:

START[=services]	LOG
STOP[=services]	SERVER
ABORT	STATUS
AUTOLOGON	VERSION
LOADKEYS	

Discussion

NS 3000/iX Network Services are composed of user services, each of which performs a specific task. These services are VT, Reverse VT, NFT, RPM, RFA, RDBA, LOOPBACK and NSSTAT. Refer to *Using NS 3000/iX Network Services* for details on Network Services.

To function, Network Services require Network Interprocess Communication (NetIPC), the user interface included with NS 3000/iX links. NetIPC is used extensively by the Network Services when processing change requests and is available for use in customer applications. It is not a service in the same sense as VT or RFA, since it consists of a set of intrinsics and associated code in the system SL and NL. NetIPC intrinsics are described in the *NetIPC 3000/XL Programmer's Reference Manual.*

The `NETCONTROL START` command must be issued before `NSCONTROL START`. This is because the `NETCONTROL` command controls the network transport subsystem, which must be initiated before the Network Services or any NetIPC application can successfully execute. NetIPC depends on the network transport to identify sockets and exchange messages. Refer to the `NETCONTROL START` command, also described in this section.

NSCONTROL ABORT

Immediately terminates all the servers and services of the Network Services.

Syntax

```
NSCONTROL ABORT
```

Parameters

ABORT	Immediately terminates all NS servers and services *without allowing existing processes to run to completion*. Useful in cases where you need to terminate Network Services immediately regardless of whether or not existing processes terminate normally. Note that STOP is the normal way to shutdown Network Services.

Discussion

There are two NSCONTROL functions that you can use to terminate the Network Services. Before using either method, be sure to warn all users that their network services are about to be closed.

- NSCONTROL STOP allows existing users to continue using the services until they finish their tasks and prevents any new users from using the services. Using NSCONTROL STOP is the recommended way to terminate the Network Services. It allows the services to terminate gracefully.
- NSCONTROL ABORT immediately terminates all the services and all the server processes. Use NSCONTROL ABORT only when you don't care about letting existing processes run to completion. Anyone using a service finds their task (REMOTE, DSCOPY, and so forth) immediately terminated.

HP recommends that you use the sequence of NSCONTROL STOP followed by NSCONTROL ABORT to ensure that you terminate all Network Services. Special situations where this may be appropriate include when the system is being prepared for software installation, or when the system needs to be taken down for maintenance. Abnormal situations can occur when an application has been incorrectly implemented. If you cannot terminate the session by any other method, use NSCONTROL ABORT to terminate all Network Services. This will clear any problems. The sequence to use prior to issuing the NSCONTROL ABORT command is shown in example 1; an abnormal situation is described in example 2.

Example 1

Issue a message to all users to stop using the Network Services. Use whatever method is appropriate for your installation. Then use the following to terminate the Network Services:

```
NSCONTROL ABORT
```

Prevents any users or programs from accessing Network Services.

Example 2

If a remote session has been terminated by the user but still shows as active on a SHOWJOB display, use ABORTJOB to terminate the session.

In the unlikely event that ABORTJOB does not work, use NSCONTROL ABORT. Be sure to follow the sequence shown in the examples before issuing the NSCONTROL ABORT command.

```
:NSCONTROL STATUS=USERS

NO CURRENT NETWORK SERVICE USERS
```

Checks that all users of the Network Services are finished.

```
:NSCONTROL ABORT
```

Terminates all users, services, and server processes.

NSCONTROL AUTOLOGON

Enables or disables the autologon feature of certain NS 3000/iX services.

Syntax

```
NSCONTROL AUTOLOGON=[{ON } [,ALL]]
                    [{OFF} [,service[,service]...]
```

Parameters

AUTOLOGON — Enables or disables the automatic logon feature available with the NFT, RFA, and RPM services. If AUTOLOGON is not enabled, users must create a remote session with the REMOTE HELLO command prior to executing these services.

- ON — Enables autologon for an NS 3000/iX service.
- OFF — Disables autologon for an NS 3000/iX service.
- ALL — Alters the autologon state for the NFT, RFA, and RPM services.

The services that allow autologon are:

- NFT — Changes autologon capability for the NFT service.
- RFA — Changes autologon capability for the RFA service.
- RPM — Changes autologon capability for the RPM service.

Defaults: ON and ALL.

Discussion

NSCONTROL AUTOLOGON allows the user the ability to disable and re-enable autologon for the NS 3000/iX services supporting this feature. Autologon is enabled at NS 3000/iX startup. NSCONTROL AUTOLOGON must be executed after the NSCONTROL START command. When the NS 3000/iX services are stopped, the autologon option resets to the default.

Disabling autologon may be important on those systems that use a logon UDC to help enforce system security. With autologon enabled (the default), a remote user can access local NFT, RFA and RPM services without executing logon UDC's. With autologon disabled, remote users must first establish a remote session with the REMOTE HELLO command, and thus execute any preset logon UDC(s), before using an NS 3000/iX service.

It is recommended that users with security logon UDCs disable autologon for all services in order to preserve the security of the system from remote users. Incoming requests attempting to use the autologon feature will fail, since a remote session cannot be established automatically.

NSCONTROL LOADKEYS

Loads the Network Services command keywords.

Syntax

```
NSCONTROL LOADKEYS
```

Parameters

LOADKEYS	Loads the Network Services command keywords from the ASCAT.NET.SYS catalog. You need to use this command only if the catalog is modified, such as for localization.

Discussion

The LOADKEYS function is only used to switch between pre-prepared ASCAT.NET.SYS catalogs. When the node is initiated, the Network Services command keywords are automatically loaded into an extra data segment from the ASCAT.NET.SYS catalog. This is done to ensure fast access to the command keywords during command parsing. However, it might be useful to have commands in the appropriate language of the installation. If so, the LOADKEYS function is used to reload the alternate catalog into the extra data segment without having to coolstart the system. Make a copy and a listing of the catalog before switching catalogs.

Example

```
:HELLO MANAGER.SYS,NET
```

Logon to the NET group in the SYS account

```
:RENAME ASCAT,ASCATOLD
```

Rename the old catalog.

```
:RENAME ASCATNEW, ASCAT
```

Substitute the new catalog for the old

```
:NSCONTROL LOADKEYS
```

Reload the catalog.

NOTE If an NSCONTROL command reports CIERR 5077, follow this example to restore the old ASCAT catalog and contact your HP representative for assistance.

NSCONTROL LOG

Enables or disables detailed event logging for the Network Services.

Syntax

```
                    [{,ALL     }        ]
                {ON } [{,RPM     } {,LOW} ]
NSCONTROL LOG={OFF} [{,ENV     } {,HIGH}]
                    [{,DSDAD   }        ]
                    [{,VTSERVER}        ]
                    [{,DSSERVER}        ]
```

Parameters

`LOG` Enables or disables NMS logging of Network Services detailed events, configured as SUB0006, CLAS0004 in the `NMCONFIG.PUB.SYS` configuration file. Detailed events are only used for troubleshooting and are normally disabled.

- `ON` Enables detailed logging of the specified Network Service modules.
- `OFF` Disables detailed logging of the specified Network Service modules.

For each Network Services software module, two levels of event logging are provided. These are HIGH, which logs all events, and LOW, the default, which logs a subset of the events, as specified below.

- `ALL` `LOW` — Logs `LOW` events for all modules.

  `HIGH` — Logs `HIGH` events for all modules.
- `RPM` `LOW` — Logs `RPMCREATE` and `RPMKILL` requests.

  `HIGH` — Same as `LOW`.
- `ENV` `LOW` — Logs environment information from `DSLINE` and `REMOTE HELLO` commands.

  `HIGH` — Same as `LOW`, plus environment table locking and use counts.
- `DSDAD` `LOW` — Logs creation and deletion of sockets, ports, and server processes.

	HIGH — Same as LOW, plus all received change requests and internal messages between DSDAD and server processes.
VTSERVER	LOW — Logs internal initialization messages between DSDAD and user processes.
	HIGH — Same as LOW, plus all received messages from other processes.
DSSERVER	LOW — Logs internal initialization messages between DSDAD and user processes.
	HIGH — Same as LOW, plus all received messages from other processes.
	Defaults: ALL and LOW

Discussion

One of the log classes defined for the Network Services is detailed event logging, which records normal Network Services events. When first started and during normal operation, the Network Services detailed event logging is disabled in order to avoid the overhead of frequent logging. Typically, detailed event logging is only enabled to investigate a specific action or series of events if required for troubleshooting.

When detailed event logging is enabled, the log messages destination is determined by the configuration of NMMGR logging subsystem 6 class 4 (SUB0006, CLAS0004). The log file is the recommended destination for detailed logging. Logging detailed events to the system console is not recommended, since the log messages tend to clutter the console screen.

Example

The example below logs the environment information from DSLINE and REMOTE HELLO commands and the change requests received by the DSDAD process. You might use this type of event logging to monitor usage of the Network Services. The destination for CLAS0004 of SUB0006 specified in the NMMGR logging configuration should be to the NM log file, not the system console.

```
:NSCONTROL LOG=ON,ENV,LOW;LOG=ON,DSDAD,HIGH
```

NSCONTROL SERVER

Alters the characteristics of the Network Services server processes.

Syntax

```
                 {servername}
NSCONTROL SERVER= {ALL       }[,minservers][,maxservers]
```

Parameters

SERVER — Dynamically alters the minimum or maximum number of servers.

serverName — Specifies the type of server for which you want to alter the available number of server processes. The servers that control the network services are:

- DSSERVER — The specified options apply to the server that controls RFA, RDBA, PTOP, and RPM. Default minserver, maxserver values are 0, 300 respectively.
- LOOPBACK — The specified options apply to the server used by the LOOPBACK services. Default minserver, maxserver values are 0, 300 respectively.
- NFT — The specified options apply to the server that controls NFT. Default minserver, maxserver values are 0, 300 respectively.
- NSSTATUS — The specified options apply to the server that controls NSSTAT (and NSTATL) services. Default minserver, maxserver values are 0, 300 respectively.
- VTSERVER — The specified options apply to the server that controls VT and REVERSE VT. Default minserver, maxserver values are 0, 300 respectively.

ALL — If you specify ALL in place of a servername, the specified options apply to all servers (NFT, DSSERVER, LOOPBACK, NSSTATUS, VTSERVER).

Default: ALL

There may be additional servers to control if other network products, such as Personal Productivity

Center, are installed. Refer to that network product's documentation to obtain the appropriate server names.

`minservers` The minimum number of servers which will be in existence at all times. This includes active and reserved servers. These servers are created immediately on the initiation of Network Services and are then kept in reserve until a change request is received. Once the change request is completed, the server is returned to reserve status. If necessary, additional servers are created immediately to fit the new minimum specified. Valid range: 0–1250; however, see the following note.

Default: 0

`maxservers` The maximum number of servers. If necessary, reserved servers will be terminated to fit the new maximum. However, a server that is in use will not be terminated until it is returned to the reserved server pool.

Limits in the number of allowed processes and internal data structures can prevent you from reaching the maximum number of servers. Valid range: 0–32767; however, see the following note.

Default: Varies by server

NOTE The total number of all active servers may not exceed 1250. The sum of all minservers must always be 1250 or less. You may specify a number greater than 1250 as one or more maxservers values, but there will never be more than a total of 1250 servers of all kinds at any one time.

Discussion

The number of server processes is controlled with the SERVER function. The maximum number of servers limits how many processes of each server type can be in existence at any time. If the servers are at the maximum limit and a new change request (such as a DSCOPY or REMOTE HELLO) is received, the request will be rejected. By setting a maximum limit, you can control the amount of process resources available for NS 3000/iX.

Because the creation and initialization of a server takes time, using reserved servers decreases the set up time for a change request. A reserved server is created ahead of time and is held in reserve until a change request is received. The minimum number of servers controls the number of reserved processes for each type of server. The number set for the minimum does not limit the number of concurrent users of the Network Services. If there are more concurrent users than the minimum number of servers specified, new users can use the Network Services, but there is a delay while the additional servers are created.

There is no simple formula for determining how many precreated servers to specify. Since each precreated server consumes one set of process resources, including process related system table entries and virtual memory for stack space, the number chosen must be a tradeoff between using system resources and allowing fast service response. The node manager needs to estimate, on the average, the number of concurrent users of each type of server. This number is used for the minimum number of servers of each type. Since the DSSERVER process is used by several services, and some of these services are active for a long time, it makes sense to allocate a larger number of DSSERVER servers than NFT, VTSERVER, LOOPBACK or NSSTATUS servers.

An alternative to allocating a greater number of DSSERVER servers is to allocate the program files NFT.NET.SYS, VTSERVER.NET.SYS, DSSERVER.NET.SYS, LOOPBACK.NET.SYS, and NSSTATUS.NET.SYS. This alternative is most advantageous for DSSERVER, where the allocation of the program file is a significant portion of the set up time. The NFT server must read keywords and messages from the NFTCAT2 catalog as well as allocate the program file when the server is created, so the performance gain is not as great as for DSSERVER.

Creating reserved servers or using the allocation alternative means that the program file is in use, just as when a program is run. Since the program file is in use, it cannot be purged, replaced, or backed up. Before any software installation, when the program files are replaced or backed up, check that the program files are not allocated and that there are no reserved servers.

Example

The following command sets the minimum number of DSSERVER processes to five and the maximum to 10. Five reserved DSSERVER processes are created immediately and are available for future change requests. The minimum number of servers, which includes both reserved and active servers, is restricted to five. When an active server is returned to the reserved pool, if there are already five reserved servers, the extra server is terminated. The maximum limit means that if there are 10 DSSERVER processes active, any new change requests will be rejected.

```
:NSCONTROL SERVER=DSSERVER,5,10
```

Example

If you execute the following command, there will be 10 server processes created for NFT, 10 for VTSERVER, 10 for DSSERVER, 10 for LOOPBACK, and 10 for NSSTATUS. Later, when users issue change requests (such as DSCOPY and REMOTE HELLO), they do not have to wait for the servers to be created. The maximum number of servers is unchanged.

```
:NSCONTROL SERVER=ALL,10
```

Example

In the following example, the node manager has chosen to allocate the program file used for the DSSERVER servers and to establish two reserved servers for NFT. To limit the system resources available, the maximum number of servers is set to 10 for both server types. In this way, performance is improved with a minimum amount of system resources used. Notice that the SERVER function can be repeated; multiple instances of NSCONTROL functions are allowed on the same command line.

```
:ALLOCATE DSSERVER.NET.SYS
:NSCONTROL SERVER=NFT,2,10;SERVER=DSSERVER,,10
:NSCONTROL STATUS=SERVERS

SERVER    MIN   MAX   DEBUG   PIN   JOBNUM   STATUS   SERVICES

LOOPBACK  0     300   OFF
NFT       2     10    OFF
                          247       RESERVED
                          187       RESERVED
DSSERVER  0     10    OFF
NSSTATUS  0     300   OFF
VTSERVER  0     300   OFF
```

NSCONTROL START

Enables the Network Services.

Syntax

```
NSCONTROL START[=service[,service]…]
```

Parameters

START[=*services*]

Enables the Network Services (VT, Reverse VT, NFT, RFA, RDBA, RPM, LOOPBACK, and NSSTAT). The first START creates the Network Services control process, called DSDAD. The optional service list (services) allows you to select which of the services are enabled for local or remote use.

Default (if the service list is omitted): enables all services for both local and remote use.

The services which allow users on remote nodes to use resources on the local node are as follows:

LOOPBACK	Allows remote users to use the loopback diagnostic server on the local node.
NFT	Allows remote users to transfer files to or from the local node using the DSCOPY command and intrinsic.
NSSTAT	Allows remote users to use the NSSTATUS intrinsic to retrieve network services information from the local node.
RFA	Allows remote users to access files on the local node, using the RFA and RDBA services.
RPM	Allows remote users to create and kill processes on the local node using the Remote Process Management service.
VT	Allows remote users to logon to a session on the local node.
VTR	Allows remote users to access local terminals using the Reverse VT service.

VTA	Allows remote users who are running the Virtual Terminal service over TCP implementations which only support the ARPA standard stream mode flow control mechanisms to log onto the local node.

The services which allow users on the local node to use resources on remote nodes are:

NFTL	Allows local users to transfer files to or from remote nodes using the DSCOPY command and intrinsics.
NSSTATL	Allows local users to use the NSSTATUS intrinsic to retrieve network services information from the local and remote nodes.
RFAL	Allows local users to open and access files and databases on remote nodes, using the RFA and RDBA services.
RPML	Allows local users to create and kill processes on the local and remote nodes using the RPM service.
VTL	Allows local users to log onto remote nodes using the REMOTE HELLO command.
VTRL	Allows local users to access terminals on remote nodes using the Reverse VT service.

Discussion

If you issue an NSCONTROL START without specifying a service list, the default is to start all services. You use the service list if you wish to select which services to start, and whether local or remote users are allowed to use the services. To allow remote users to use VT, VTR, VTA, NFT, RFA/RDBA, NSSTAT, LOOPBACK, and RPM on your local node, you must START the appropriate remote services. Additionally, if you wish to allow local users to use VT, NFT, RFA/RDBA, RPM, and NSSTAT to remote nodes, you must START the appropriate local services.

You must issue the NETCONTROL START command before the NSCONTROL START command. This is because the Network Services depend on the network transport subsystem. Refer to the NETCONTROL START command for more information.

Example 1

Example 1 shows the command sequence necessary to start the Network Services. Enter the NETCONTROL START command to initiate the network transport before the NSCONTROL START command, as shown in the example. Issuing the NSCONTROL START creates the DSDAD process and starts all the user services.

To successfully initialize a node, the commands must be issued in the order specified. At least one of the required NETCONTROL START commands must be issued first, before the NSCONTROL START command.

```
:NETCONTROL START;NET=LAN1
:NSCONTROL START
```

Example 2

For security reasons, the node manager for this node has decided to restrict the Network Services to outgoing only. The command shown in example 2 enables users on the local node to use resources on remote nodes. The reverse is not true. Users on remote nodes are not allowed to logon or use any of the services on the local node. The status display shows all the local services enabled and all the remote services disabled.

```
:NSCONTROL START=VTL,VTRL,NFTL,RFAL,RPML
VTL NETWORK SERVICE STARTED.
VTRL NETWORK SERVICE STARTED.
NFTL NETWORK SERVICE STARTED.
RFAL NETWORK SERVICE STARTED.
RPML NETWORK SERVICE STARTED.

:NSCONTROL STATUS=SERVICES

SERVICE    TYPE    STARTED  SERVER     DESCRIPTION

VTA        REMOTE     NO    VTSERVER   INCOMING STREAM MODE VIRTUAL TERMINAL
NSSTATL    LOCAL      NO    NSSTATUS   OUTGOING NSSTATUS SERVICE
NSSTAT     REMOTE     NO    NSSTATUS   INCOMING NSSTATUS SERVICE
LOOPBACK   REMOTE     NO    LOOPBACK   INCOMING LOOPBACK SERVICE
RPML       LOCAL      YES    DSSERVER   OUTGOING REMOTE PROCESS MANAGEMENT
RPM        REMOTE     NO    DSSERVER   INCOMING REMOTE PROCESS MANAGEMENT
PTOPL      LOCAL      NO    DSSERVER   OUTGOING PROGRAM-TO-PROGRAM COMMUNICATION
PTOP       REMOTE     NO    DSSERVER   INCOMING PROGRAM-TO-PROGRAM COMMUNICATION
RFAL       LOCAL      YES    DSSERVER   OUTGOING REMOTE FILE ACCESS
RFA        REMOTE     NO    RASERVER   INCOMING REMOTE FILE ACCESS
NFTL       LOCAL      YES    NFT        OUTGOING NETWORK FILE TRANSFER
NFT        REMOTE     NO    NFT        INCOMING NETWORK FILE TRANSFER
VTRL       LOCAL      YES    VTSERVER   OUTGOING REVERSE VIRTUAL TERMINAL
VTR        REMOTE     NO    VTSERVER   INCOMING REVERSE VIRTUAL TERMINAL
VTL        LOCAL      YES    VTSERVER   OUTGOING VIRTUAL TERMINAL
VT         REMOTE     NO    VTSERVER   INCOMING VIRTUAL TERMINAL
```

Example 3

The network transport must be initialized before you can issue the NSCONTROL START command. If not, the error messages shown in example 3 are displayed.

```
:NSCONTROL START

TRANSPORT NOT INITIALIZED (DSERR 644)
INVALID CONTROL OPTION (CIERR 5062)
```

NSCONTROL STATUS

Displays information about the Network Services.

Syntax

```
                 [=USERS    ]
                 [=SERVICES]
NSCONTROL STATUS [=SERVERS  ]
                 [=ALL      ]
                 [=SUMMARY  ]
```

Parameters

STATUS — Displays information about the Network Services. Can be used to check if the Network Services were successfully initiated, or to check on the current status using the following parameters:

USERS	Displays the sessions on the node that are associated with the Network Services.
SERVICES	Displays information about the services.
SERVERS	Displays information about the servers.
SUMMARY	Displays a summary of the information about services, servers, and users.
ALL	Displays all available information about services, servers, and users.

You can qualify the STATUS function with one parameter or with a list of parameters separated by commas.

Default: ALL

Discussion

This function displays information on those local sessions that were created by a DSLINE and REMOTE HELLO and on those remote sessions that were created by a REMOTE HELLO. The STATUS display does not list information on either local sessions that are using DSCOPY without a REMOTE HELLO or temporary remote sessions created by NFT, RFA, or RPM.

The following examples show the information provided by the STATUS function of the NSCONTROL command.

Example 1

The following example shows the status of the Network Services. Local means the service gives local users access to remote resources; remote means the service gives remote users access to local resources. Server indicates the type of server, NFT or DSSERVER, used for the service. For this example, all the services were started as indicated by YES in the STARTED column of the display. A NO in that column would indicate that the service was not started. You can use the STATUS display to verify whether each individual service is started or not, and whether it is available for local or remote use. This is helpful when using the optional services list of the NSCONTROL START and STOP functions.

```
:NSCONTROL STATUS=SERVICES

SERVICE    TYPE   STARTED   SERVER     DESCRIPTION

VTA        REMOTE    YES    VTSERVER   INCOMING STREAM MODE VIRTUAL TERMINAL
NSSTATL    LOCAL     YES    NSSTATUS   OUTGOING NSSTATUS SERVICE
NSSTAT     REMOTE    YES    NSSTATUS   INCOMING NSSTATUS SERVICE
LOOPBACK   REMOTE    YES    LOOPBACK   INCOMING LOOPBACK SERVICE
RPML       LOCAL     YES    DSSERVER   OUTGOING REMOTE PROCESS MANAGEMENT
RPM        REMOTE    YES    DSSERVER   INCOMING REMOTE PROCESS MANAGEMENT
PTOPL      LOCAL     YES    DSSERVER   OUTGOING PROGRAM-TO-PROGRAM COMMUNICATION
PTOP       REMOTE    YES    DSSERVER   INCOMING PROGRAM-TO-PROGRAM COMMUNICATION
RFAL       LOCAL     YES    DSSERVER   OUTGOING REMOTE FILE ACCESS
RFA        REMOTE    YES    RASERVER   INCOMING REMOTE FILE ACCESS
NFTL       LOCAL     YES    NFT        OUTGOING NETWORK FILE TRANSFER
NFT        REMOTE    YES    NFT        INCOMING NETWORK FILE TRANSFER
VTRL       LOCAL     YES    VTSERVER   OUTGOING REVERSE VIRTUAL TERMINAL
VTR        REMOTE    YES    VTSERVER   INCOMING REVERSE VIRTUAL TERMINAL
VTL        LOCAL     YES    VTSERVER   OUTGOING VIRTUAL TERMINAL
VT         REMOTE    YES    VTSERVER   INCOMING VIRTUAL TERMINAL
```

Example 2

Example 2 shows the status of the servers. Here the minimum number of NFT servers is 0 and the maximum is 300 (the defaults). There are no NFT servers created. The minimum number of VTSERVERS is 6 and the maximum is 300. One, with process ID number (PIN) 50, is active, being used for the VT service with session #S1. The other five are not being used but are in reserve.

```
:NSCONTROL STATUS=SERVERS

SERVER     MIN   MAX   ACTIVE    RESERVED   DEBUG   PIN   JOBNUM   STATUS

RASERVER     0   300         0          0   OFF
NSSTATUS     0   300         0          0   OFF
LOOPBACK     0   300         0          0   OFF
VTSERVER     6   300         1          5   OFF
                                                  50   #S1     ACTIVE
```

```
                                                    51          RESERVED
                                                    49          RESERVED
                                                    41          RESERVED
                                                    58          RESERVED
                                                    57          RESERVED
NFT          0  300         0          0  OFF
DSSERVER     0  300         0          0  OFF

TOTAL NUMBER OF ACTIVE SERVERS:      1
TOTAL NUMBER OF RESERVED SERVERS:    5
TOTAL NUMBER OF SERVERS:             6
```

Example 3

In example 3, assume that a user has entered the following commands on NODE1:

```
:HELLO MANAGER.SYS
:DSLINE NODE2
:REMOTE HELLO MGR.TELESUP
```

The result on NODE1 is:

```
:NSCONTROL STATUS=USERS

JOBNUM    SESSION  TYPE         USER.ACCOUNT
          ID         SERVICES     NODENAME

#S1       #060507  LOCAL        MANAGER.SYS
          #031237    VT           NODE2.DOMAIN.ORGANIZATION

TOTAL NUMBER OF LOCAL NS USERS:      1
TOTAL NUMBER OF REMOTE NS USERS:     0
TOTAL NUMBER OF NS USERS:            1
```

and on NODE2:

```
:NSCONTROL STATUS=USERS

JOBNUM    SESSION  TYPE         USER.ACCOUNT
          ID         SERVICES     NODENAME

#S3       #031237  REMOTE       MGR.TELESUP
          #060507    ORIGIN       NODE1.DOMAIN.ORGANIZATION

TOTAL NUMBER OF LOCAL NS USERS:      0
TOTAL NUMBER OF REMOTE NS USERS:     1
TOTAL NUMBER OF NS USERS:            1
```

The display on NODE2 shows the remote session for MGR.TELESUP from the REMOTE HELLO on NODE1. As illustrated in example 3, the session IDs can be used to match up the local and remote sessions. The local session on NODE1, with ID #060500, is the origin of the remote session on NODE2, with ID #031237.

Example 4

In the following example, the Network Services have not been started (no NSCONTROL START has not been issued). The system response to the NSCONTROL STATUS=USERS,SERVICES command shows that there are no Network Services users and no Network Services currently active.

```
:NSCONTROL STATUS=USERS,SERVICES

NO CURRENT NETWORK SERVICE USERS

NO NETWORK SERVICES ARE CURRENTLY ACTIVE
```

Example 5

Example 5 shows the brief summary of users, services, and servers information. This is an abbreviated display of STATUS=ALL.

```
:NSCONTROL STATUS=SUMMARY

TOTAL NUMBER OF LOCAL NS USERS:          1
TOTAL NUMBER OF REMOTE NS USERS:         0
TOTAL NUMBER OF NS USERS:                1

      OUTGOING SERVICES                        INCOMING SERVICES
SERVICE     STARTED      FEATURES       SERVICE     STARTED      FEATURES

NSSTATL       YES                       VTA           YES
RPML          YES                       NSSTAT        YES
PTOPL         YES                       LOOPBACK      YES
RFAL          YES                       RPM           YES  AUTOLOGON OFF
NFTL          YES                       PTOP          YES
VTRL          YES                       RFA           YES  AUTOLOGON ON
VTL           YES                       NFT           YES  AUTOLOGON ON
                                        VTR           YES
                                        VT            YES

SERVER            MIN  MAX  ACTIVE      RESERVED  DEBUG

RASERVER            0  300       0            0   OFF
NSSTATUS            0  300       0            0   OFF
LOOPBACK            0  300       0            0   OFF
VTSERVER            6  300       1            5   OFF
NFT                 0  300       0            0   OFF
DSSERVER            0  300       0            0   OFF

TOTAL NUMBER OF ACTIVE SERVERS:        1
TOTAL NUMBER OF RESERVED SERVERS:      5
TOTAL NUMBER OF SERVERS:               6
```

NSCONTROL STOP

Terminates Network Services subsystem.

Syntax

```
NSCONTROL STOP[=service[,service]…]
```

Parameters

STOP=*services*

Terminates the Network Services subsystem. STOP executes a “graceful” shutdown of Network Services. Existing users of the service can continue until they complete their NS activity, but new users are prevented from using the services. The optional service list (services) allows you to select which of the services are disabled for local or remote use. When all Network Services are stopped, the DSDAD process will terminate.

Default (if the service list is omitted): terminates all services for both local and remote use.

The services list is the same as for the START function, except that the specified services are stopped, not started.

Specifying the following services prevents users on remote nodes from using resources on the local node:

LOOPBACK	Prevents remote users from using the loopback diagnostic server on the local node.
NFT	Prevents remote users from transferring files to or from the local node using the DSCOPY command and intrinsic.
NSSTAT	Prevents remote users from using the NSSTATUS intrinsic to retrieve network services information from the local node.
RFA	Prevents remote users from accessing files on the local node.
RPM	Prevents remote users from creating and killing processes on the local node using the Remote Process Management

service.

VT	Prevents remote users from logging onto the local node using the REMOTE HELLO command.
VTR	Prevents remote users from accessing local terminals using the Reverse VT service.
VTA	Prevents remote users who are running the Virtual Terminal service over TCP implementations which only support the ARPA standard stream mode flow control mechanisms to log onto the local node.

Specifying the following services prevents users on the local node from using resources on remote nodes:

NFTL	Prevents local users from transferring files to or from remote nodes using the DSCOPY command and intrinsics.
NSSTATL	Prevents local users from using the NSSTATUS intrinsic to retrieve network services information from the local and remote nodes.
RFAL	Prevents local users from opening and accessing files and databases on remote nodes using the RFA and RDBA services.
RPML	Prevents local users from creating and killing processes on the local and remote nodes using the RPM service.
VTL	Prevents local users from logging onto remote nodes using the REMOTE HELLO command.
VTRL	Prevents local users from accessing terminals on remote nodes using the Reverse VT service.

Discussion

NSCONTROL STOP is the normal way to shut down the Network Services. It allows existing users to continue using the services until they finish their tasks, but prevents any new users from using the services. The ABORT function, on the other hand, immediately terminates all the services and all the server processes. Anyone using a service will find their task (DSCOPY, for example) immediately terminated. See the discussion of NSCONTROL ABORT.

Example 1

Example 1 shows NSCONTROL STOP without the service list. All Network Services are stopped. Any active servers are allowed to continue until finished with the current task, at which point they are terminated. No new change requests are accepted. When all the servers and services are stopped, the DSDAD process terminates. In the example, an NSCONTROL ABORT command is issued after the NSCONTROL STOP command to make sure all Network Services activity is stopped.

```
:NSCONTROL STOP   {users and servers allowed to finish}
:NSCONTROL ABORT  {terminates all network services activity}
:NETCONTROL STOP  {terminates all network transport activity}
```

Example 2

The NSCONTROL STOP=VT,VTA,VTR command shown in the following example stops the VT, VTA, and Reverse VT services. This prevents remote users from logging on to the local node using REMOTE HELLO and from opening local terminals using Reverse VT. If there are any other active Network Services, they remain available.

```
:NSCONTROL STOP=VT,VTA,VTR
```

NSCONTROL VERSION

Displays the version numbers for the Network Services software modules and the overall subsystem version.

Syntax

```
NSCONTROL VERSION[=MOD]
```

Parameters

`VERSION[=MOD]`	Displays the overall version of the Network Services. If qualified with the MOD keyword, displays the version of each of the software modules of the Network Services as well as the overall version.

Discussion

The software modules of all HP products have a version identification number which includes the version, update, and fix level of the software module. The `VERSION` function of the `NSCONTROL` command allows you to check the version numbers of the Network Services software modules to ensure that they are compatible and up-to-date. The display is the same as that for NMMAINT except that only the Network Services subsystem is displayed.

Example 1

In example 1, the information that is provided includes the version number of the Network Services.

```
:NSCONTROL VERSION

NS3000/iX Transport 32098-20033  overall version = B.05.00
```

Example 2

To see the version numbers of the individual modules, you would specify the command using the MOD keyword. The result would be a display similar to the one shown in example 2:

```
:NSCONTROL VERSION=MOD

Network Services individual module versions:

NM Program:      DSDAD.NET.SYS         Version:  B0010005
SL procedure:    ASCXVERS              Version:  B0010006
SL procedure:    ASBUFVERS             Version:  B0010000
SL procedure:    ASENVVERS             Version:  B0010005
SL procedure:    DSUTILVERS            Version:  B0010005
SL procedure:    SUBSYS6FMTVERS        Version:  B0010001
Catalog file:    ASCAT.NET.SYS         Version:  B0010001
SL procedure:    VTSRVTVER             Version:  B0010000
NL procedure:    VTS_LDMVER            Version:  B0010003
NL procedure:    VTS_UTILVER           Version:  B0010003
CM Program:      LOOPBACK.NET.SYS      Version:  B0010000
CM Program:      LOOPINIT.NET.SYS      Version:  B0010000
CM Program:      NSSTATUS.NET.SYS      Version:  B0010002
NL procedure:    NSSTATUSNMVERS        Version:  B0010001
NL procedure:    NSINFONMVERS          Version:  B0010002
CM Program:      CONFPROG.NET.SYS      Version:  B0010000
CM Program:      MASTMAKE.NET.SYS      Version:  B0010000
NL procedure:    VTS_SMVER             Version:  B0010003
NL procedure:    NSUTILNMVERS          Version:  B0010003
NL procedure:    ASCXNMVERS            Version:  B0010002
NL procedure:    ASENVNMVERS           Version:  B0010002
NM Program:      RASERVER.NET.SYS      Version:  B0010002
NM Program:      VTSERVER.NET.SYS      Version:  B0010004
CM Program:      DSSERVER.NET.SYS      Version:  B0010000
SL procedure:    ASRFAVERS             Version:  B0010002
SL procedure:    ASPTOPVERS            Version:  B0010001
CM Program:      NFT.NET.SYS           Version:  B0010013
NL procedure:    NFTNMVERS             Version:  B0010001
Catalog file:    NFTCAT2.NET.SYS       Version:  B0010001
SL procedure:    ASRPMVERS             Version:  B0010001
NL procedure:    RPMNMVERS             Version:  B0010001
CM Program:      RPMDAD.NET.SYS        Version:  B0010003
NL procedure:    RFANMVERS             Version:  B0010001

Network Services overall subsystem version = B.00.10
```

RESUMENMLOG

Resumes logging after a recoverable error.

Syntax

```
RESUMENMLOG
```

Use

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capabilities?		NM

Discussion

RESUMENMLOG causes the resumption of logging to the NM disk log file upon the correction of a recoverable I/O error.

For example assume that the system is on line, NM logging is enabled, and a recoverable error occurs on NMLG file number 104. The error is corrected and the RESUMENMLOG command is entered. The following message is then displayed on the system console:

```
NMLG FILE NUMBER nnnn.  NM LOGGING RESUMED

NMLG FILE NUMBER nnnn ON
```

Refer to the *NS 3000/iX Error Messages Reference Manual* for more information on recoverable errors.

SHOWNMLOG

Displays the number and available space of the log file.

Syntax

```
SHOWNMLOG
```

Use

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capabilities		NM

Discussion

SHOWNMLOG displays the number of the current NMLG file and the percentage of available file space currently used.

The information appears in the following format:

NMLG FILE NUMBER *nnnn* IS *mm%* FULL

where *nnnn* is the NMLG file number and *mm* is the percentage of file space used.

If network logging is disabled due to an irrecoverable error, NMS displays the following message explaining the cause. The manager will have to do a warm or cool start to bring up the system again.

NMLG FILE NUMBER *nnnn* ERROR *#nn*. NM LOGGING STOPPED. (NMCNERR 36)

If network logging is enabled but currently suspended due to a recoverable error, NMS displays the following messages explaining the cause. Once the error is corrected, the manager can then issue the RESUMENMLOG command explained in this section.

NMLG FILE NUMBER *nnnn* IS *mm%* FULL

NMLG FILE NUMBER *nnnn* ERROR *#mm*. NMLOGGING SUSPENDED.(NMCNERR 38)

SWITCHNMLOG

Closes the current log file and creates and opens a new one.

Syntax

```
             [UPDATE    ]
SWITCHNMLOG [filenumber]
```

Use

```
Available    In Session?          YES

             In Job?              YES

             In Break?            YES

             Programmatically?    YES

Breakable?                        NO

Capabilities?                     NM
```

Parameters

UPDATE — Allows you to update logging configuration for all subsystems actively logging without stopping transport. To change the logging configuration, use the node management configurator (NMMGR). Refer to the *NS 3000/iX Screens Reference Manual* or to the *HP e3000/iX Network Planning and Configuration Guide* for more information on configuring logging. Once changes are made, issue the SWITCHNMLOG UPDATE command so that the changes take effect.

The UPDATE option may not be used with the *filenumber* option.

filenumber — Switches the log file to a file with the specified number. The value for *filenumber* must be an integer from 0 to 9999. For example,

```
SWITCHNMLOG 10
```

makes NMLG0010 the current log file. If you specify a number that is already being used, then the next available consecutive number is used.

The *filenumber* option may not be used with the UPDATE option.

Discussion

SWITCHNMLOG closes the current NMLG file and creates and opens a new one. When you enter SWITCHNMLOG, NMS displays the message:

NMLG FILE NUMBER *nnnn* IS *mm%* FULL

NMLG FILE NUMBER *pppp* ON

where *nnnn* is the previous NMLG FILE number, *mm* is the percentage of file space used, and *pppp* is the newly opened file numbered one more than the last file number.

If network logging is disabled due to an irrecoverable error when SWITCHNMLOG is entered, NMS displays the following message explaining the cause. The system will need to be brought back up with a warm or cool start.

NMLG FILE NUMBER *nnnn* ERROR *#nn*. NM LOGGING STOPPED. (NMCNERR 36)

If network logging is enabled but currently suspended due to a recoverable error, NMS displays the following message explaining the cause. When the problem is corrected, the manager can issue the RESUMENMLOG command.

NMLG FILE NUMBER *nnnn* ERROR *#nn*. NM LOGGING SUSPENDED. (NMCNERR 38)

A LINKCONTROL Command

This appendix defines the fields output by the LINKCONTROL STATUS command and its associated parameters. The LINKCONTROL STATUS command enables you to obtain link configuration and statistical data which you can use for monitoring and debugging the link. This command has several parameters, each of which provides different configuration or statistical data. The parameters described in this appendix are as follows:

- LINKSTATE
- CONFIGURATION
- STATISTICS
- ALL
- RESET

The STATUS=DIAGSTAT command returns information intended for HP diagnostic use. Its output is not explained in this manual. After issuing this command you should send the output to your HP representative.

The LINKCONTROL command displays specific information relating to the type of link that is monitored. The LINKCONTROL command can be used to obtain information about the following types of NS 3000/iX links:

- NS 3000/iX LAP-B links
- NS 3000/iX IEEE 802.3 or LAN links
- NS 3000/iX IEEE 802.5/Token Ring links
- NS 3000/iX Fiber Distributed Data Interface (FDDI) links
- NS 3000/iX 100VG-AnyLAN links
- NS 3000/iX 100Base-T links
- NS 3000/iX LAPBMUX links

The LINKCONTROL command does not work on an X.25 link because the card is located in the DTC. For equivalent functionality, use the OpenView DTC manager.

NS 3000/iX LAP-B Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX LAP-B Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   LAPB20   Linktype:   LAPB   Linkstate:   CONNECTED LEVEL 2
```

Linkname — The Linkname field specifies the name of the link.

Linktype — This Linktype specifies the type of link, such as LAP-B or IEEE 802.3, that is being monitored.

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Not connected.
- Connected level 1.
- Connected level 2.
- Connecting level 1.
- Connecting level 2.
- Disconnecting level 1.
- Disconnecting level 2.

In this example, the current state of the LAPB link named LAPB20 is CONNECTED LEVEL 2.

The *LINKSTATE* parameter fields are displayed whenever you enter the LINKCONTROL Status command, regardless of which other parameters are specified.

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for LAP-B links displays the *LINKSTATE* parameter fields and many additional fields. These additional fields display information that is related to the link configuration and which, except for the Cable Type parameter, are input through the NMMGR configuration program.

The following is an example of the data that is displayed when you issue the `LINKCONTROL linkname;STATUS=CONFIGURATION` command:

```
Physical Path              24
Phone Number            9D1D40898765432
Modullo Count               8           Cable Type       RS232
Local Mode                HP-HP         Buffer Size                1024 Bytes
LAPB Parm K                 7           Connect Timeout            60 sec
LAPB Parm T1              300 hsec      Local Timeout              900 sec
LAPB Parm N2                20          Transmission Speed         64000 bps
```

Phone Number — The Phone Number field displays the current automatic dial phone number, as specified in the NMMGR configuration file.

Modulo Count — The Modulo Count field displays the maximum frame sequence number allowable for any frame in the network. This field can be set to a value of 8 or 128, meaning that frames are numbered from either 0 through 7, or 0 through 127.

Cable Type — The Cable Type field displays the cable type that is currently connected to the PSI card. This data is provided by the PSI card and cannot be configured through NMMGR. The possible values for this field are as follows:

- RS232
- V35
- Modem Eliminator
- RS366
- RS449
- Loopback
- None

Local Mode — The Local Mode field displays the value specified for local mode in the NMMGR link screen. The possible values for this field are as follows:

- HP–HP, for connection to another HP device configured as HP Point-to-Point.
- DTE, for connection to a device that is configured as a DCE.
- DCE, for connection to a device that is configured as a DTE.

Buffer Size — The Buffer Size field displays the current buffer size. This value will be equal to the buffer size configured in the NMMGR Link screen plus 4 bytes of overhead that is added by the level 3 protocol.

LAPB Parm K — The LAPB Parm K field displays the configured number of unacknowledged frames that are allowed in the network at any given time. For example, if this value is set to seven (7) for a node, and that node transmits seven (7) packet frames onto the network, it cannot transmit another frame until one or more of the transmitted frames are acknowledged.

Connect Timeout — The Connect Timeout field displays the current logical link level 2 connection timeout. The Connect Timeout parameter sets the amount of time a node will wait for a logical connection to a remote node to be established. If this timer expires, the node aborts the connection attempt. The abort process can take several additional seconds.

LAPB Parm T1 — The LAPB Parm T1 field displays the current value of the T1 timer. The T1 timer waits the specified number of hundredths-of-seconds for a particular frame to be acknowledged. A frame that is transmitted, but not acknowledged, before the T1 timer expires, is retransmitted.

Local Timeout — The Local Timeout field displays the value specified for the Local timer. This timer, also called a heartbeat timer, is used to monitor whether the system and/or the PSI card are functioning. The PSI card, and the system, transmit a signal, called a heartbeat signal, to each other on a specified schedule. If, for example, the heartbeat does not arrive at the PSI card from the system (or vice-versa), the card or system waits the number of seconds specified by this field. If no heartbeat arrives before this timer expires, the link is dropped. You can determine whether the PSI card or the system failed by checking to see which device is still active. The default for this field is 60 seconds and it is recommended that you do not change the value of this field. The PSI always waits 20 seconds longer than the system waits before it drops the link.

LAPB Parm N2 — The LAPB Parm N2 field displays the maximum number of times a frame is retransmitted after the LAPB Parm T1 expires. The frame is retransmitted at the LAPB Parm T1 interval for the number of times specified in this field. When this count is depleted, the frame is retransmitted at 20 second intervals. If no response is received after these transmissions, the link is brought down.

A node that is configured with the value specified, will attempt to retransmit an unacknowledged frame a maximum of 20 times at T1 intervals.

Transmission Speed — The Transmission Speed field displays the current transfer rate, or clocking, configured for the node in the NMMGR Link screen. If modems are used, the modems will control the clocking. The PSI card transmits at the clocking setting of the modem and ignores the value configured in this field.

STATISTICS Parameter Fields

The *STATISTICS* parameter for LAP-B links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed with this parameter. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=STATISTICS command:

```
Connection Duration       23:25:01     Tracing                        OFF
Data Bytes Sent              63650     Data Bytes Received          62300
Overhead Bytes Sent           8592     Overhead Bytes Receivedf      8550
Total Frames Sent             1430     Total Frames Received         1425
Data Frames Sent              1253     Data Frames Received          1246
Aborted Frames Sent          62650     Aborted Received                 0
DSR Losses                       0     Oversized Frames Received        0
CTS Carrier Losses               0     Receive Overruns                 0
DCD Carrier Losses               0     CRC Errors                       0
                                       Statistics Resets                0
```

Connect Duration — The Connect Duration field displays the length of time that the current logical (level 2) connection has existed. If there is no active connection, this field displays the length of time that the most recently established connection was in existence. This field is reset with each new connection.

Tracing — The Tracing field specifies whether tracing is currently enabled or disabled.

Data Bytes Sent — The Data Bytes Sent field displays the number of bytes that have been transmitted in the data portion of all data frames.

Data Bytes Received — The Data Bytes Received field displays the number of bytes that have been received in the data portion of all data frames.

Overhead Bytes Sent — The Overhead Bytes Sent field displays the total number of flags, level 2 address and control bytes, and frame check sequence (FCS) bytes transmitted. This value should be equal to six times the total number of frames transmitted.

Overhead Bytes Received — The Overhead Bytes Received field displays the total number of flags, level 2 address and control bytes, and FCS bytes received. This value should be equal to six times the total number of frames transmitted.

Total Frames Sent — The Total Frames Sent field displays the total number of frames transmitted.

Total Frames Received — The Total Frames Received field displays the total number of frames received.

Data Frames Sent — The Data Frames Sent field displays the total number of transmitted data frames.

Data Frames Received — The Data Frames Received field displays the total number of received data frames.

Aborted Frames Sent — The Aborted Frames Sent field displays the number of frames which were aborted before they were received. Normally, this number should be quite low (below 3% of the total number of frames sent). A large number could point to a noisy line or a weak or bad clock signal sent by a modem. If this value becomes larger than 3% of the total number of frames sent, and you feel that network performance is being affected, contact your HP representative.

Aborted Frames Received — The Aborted Frames Received field displays the number of frames which were aborted after they were received. Normally, this number should be quite low (below 3% of the total number of frames received). A large number could point to a noisy line or a weak or bad clock signal sent by a modem. If this value becomes larger than 3% of the total number of frames received, and you feel that network performance is being affected, contact your HP representative.

DSR Losses — The DSR Losses field displays the number of times the PSI detected a temporary loss of the Data Set Ready (DSR) signal on the cable. On most lines, this will remain at a value of zero, although some modems will periodically drop signals for very short intervals.

Oversized Frames Received — The Oversized Frames Received field displays the number of frames received that exceed the maximum configured buffer size as configured in the Link screen of NMMGR. A number other than zero in this field indicates that the remote buffer size configuration is greater than the local buffer size configuration. One or both configurations must be modified so that the two buffer sizes are identical.

CTS Carrier Losses — The CTS Carrier Losses field displays the number of times the PSI detected a temporary loss of the Clear to Send (CTS) signal on the cable. On most lines, this will remain at a value of zero, although some modems will periodically drop signals for very short intervals.

Receive Overruns — The Receive Overruns field displays the number of times that the PSI card had to discard a frame because the PSI card could not process the data as quickly as it arrived. A number in this field that is greater than 3% of the total number of received frames indicates a possible problem with the PSI card. If this number continues to increase, contact your HP representative.

DCD Carrier Losses — The DCD Carrier Losses field displays the number of times the PSI detected a temporary loss of the Data Carrier Detect (DCD) signal on the cable. On most lines, this will remain at a value of zero, although some modems will periodically drop signals for very short intervals.

CRC Errors — The CRC Errors field displays the number of frames that were received with a bad Cyclic Redundancy Check (CRC) checksum. A large number (greater than 1% of the total number of packets) indicates that a problem may exist in the connection between the PSI and the modem, or between the two modems.

Statistics Resets — The Statistics Resets field displays the number of times that the statistics buffer (which contains the values for all of the aforementioned fields) was reset. This value is reset each time the link is restarted.

RESET Parameter Fields

The *RESET* parameter for LAP-B links resets all of the accumulated statistics for the links. This command also displays all of the *LINKSTATE*, *CONFIGURATION*, and *STATISTICS* parameter fields. Refer to the *STATISTICS* parameter for a description of the displayed fields.

ALL Parameter Fields

The *ALL* parameter for LAP-B links displays all of the *LINKSTATE*, *CONFIGURATION*, and *STATISTICS* parameter fields. This is an example of the *ALL* parameter output:

```
Linkname:   LAPB20   Linktype:   LAPB   Linkstate:   CONNECTED LEVEL 2

Physical Path      24
Phone Number   9D1D40898765432
Modulo Count       8             Cable Type              RS232
Local Mode       HP-HP           Buffer Size                1024 Bytes

LAPB Parm K          7           Connect Timeout              60 sec
LAPB Parm T1       300 hsec      Local Timeout               900 sec
LAPB Parm N2       20            Transmission Speed        64000 bps

Connection Duration  23:25:01    Tracing                     OFF
Data Bytes Sent         62650    Data Bytes Received       62300
Overhead Bytes Sent      8592    Overhead Bytes Received    8550
Total Frames Sent        1430    Total Frames Received      1425
Data Frames Sent         1253    Data Frames Received       1246
Aborted Frames Sent     62650    Aborted Received              0
DSR Losses                  0    Oversized Frames Received     0
CTS Carrier Losses          0    Receive Overruns              0
DED Carrier Losses          0    CRC Errors                    0
                                 Statistics Resets             0
```

NS 3000/iX LAN Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX LAN Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   SYSLINK   Linktype:   IEEE802.3   Linkstate:   CONNECTED
```

Linkname — The Linkname field specifies the name of the link.

Linktype — The Linktype field specifies the type of link, such as LAP-B or IEEE 802.3, that is being monitored.

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Connected
- Not connected

NOTE Some of the parameter descriptions that follow vary according to whether your LAN card is a CIO card or an NIO card.

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for IEEE 802.3 links displays several fields in addition to the LINKSTATE parameter field. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=CONFIGURATION command:

```
Physical Path:                  4.6
Inbound Buffer Size:            1536 (CIO only)
Inbound Number of Buffers:      64 (CIO only)
Inbound Buffers Available:      41 (CIO only)
Current Station Address:        08-00-09-02-3D-9B
Default Station Address:        08-00-09-02-3D-9B
Current Receive Filter:         bad(0) multi(1) broad(1) any(0)
Current Multicast Addresses:    09-00-09-00-00-01
```

Physical Path — The Physical Path field displays the current physical path for the LAN card as specified in the NMMGR configuration file.

Inbound Buffer Size — The Inbound Buffer Size field displays the current size of the receive buffer that are configured for this system

through NMMGR. This field will not print for NIO cards, since there may be multiple inbound buffer sizes.

Inbound Number of Buffers — The Inbound Number of Buffers field displays the number of receive buffers that are configured for this system through NMMGR. This field will not print for NIO cards.

Inbound Buffers Available — The Inbound Buffers Available field displays the number of unused or unassigned Inbound Buffers that are available to this system. This field will not print for NIO cards.

Current Station Address — The Current Station Address field is a display of the six (6) byte address to which the node is configured to respond. This address is used whenever frames are sent to the network media. The default station address is used unless it is overridden in the NMMGR link configuration screen. If this field is changed, then the station address of this node is changed. Make sure that you note this new address in the system manager log.

Default Station Address — The Default Station Address field is the default value for the Current Station Address described above. The default station address is determined by the specific LAN card. It is also printed on the LAN card. If the card is changed for any reason, the Default Station Address of this node will change.

Current Receive Filter — The Current Receive Filter field has a current value. The current value is currently used by the LAN card.

Receive Filter bad ( ) — The Receive Filter bad ( ) field is either enabled (1) or disabled (0). When enabled, any bad frames that are received by the LAN are passed to the driver. When disabled, bad frames are discarded. Any bad frames are counted in the statistics.

Receive Filter multi ( ) — The Receive Filter multi ( ) field is either enabled (1) or disabled (0). When enabled, you can specify a list of specified multicast frames to be received by the LAN hardware card. The list can contain up to 64 multicast addresses to be downloaded to the LAN and is displayed when this field is entered.

Receive Filter broad ( ) — The Receive Filter broad ( ) field is either enabled (1) or disabled (0). When enabled, the LAN card receives frames sent to the broadcast address.

Receive Filter any ( ) — The Receive Filter any ( ) field is either enabled (1) or disabled (0). When enabled, the LAN card attempts to receive all frames from the network media. When disabled, only those frames sent to the LAN card are received.

Receive Filter k_pckts() (NIO card only) — The Receive Filter k_pckts() filed is either enabled (1) or disabled (0). When enabled, the LAN card keeps frames received from the network media, even if no buffers are currently posted to the card. If this option is not enabled, the frames will be dropped.

Receive Filter x_pckts() (NIO card only) — The Receive Filter x_pckts() field is either enabled (1) or disabled (0). When enabled, any XID or TEST commands sent to DSAP 0 will be responded to by the driver, and not the card.

Current Multicast Addresses — The Current Multicast Addresses field contains a list of all multicast addresses to which the LAN card responds. The default multicast address list contains no addresses. If no multicast addresses are specified, the following message is printed:

```
Current multicast address list is empty
```

Multicast addresses are configured through NMMGR. The maximum number of multicast addresses allowed is 16. The meanings of the following specific multicast addresses are as follows:

`09-00-09-00-00-01`	Probe address
`09-00-09-00-00-02`	Second probe address
`09-00-09-00-00-03`	LAN analysis (LANDAD)
`09-00-09-00-00-04`	DTC boot address

STATISTICS Parameter Fields

The *STATISTICS* parameter for IEEE 802.3 links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. The following is an example of the data that is displayed when you issue the *LINKCONTROL linkname;STATUS=STATISTICS* command:

```
Transmits no error          1     Receives no error      343
Transmits error             0     Receives error           0
Out of TX bufs              0     Out of Rx bufs           0
Transmits deferred          0     Carrier losses           0
Transmits 1 retry           0     Reflectometer            0
Transmits >1 retry          0     CRC errors               0
Transmits 16 collisions     0     Whole byte errors        0
Transmits late collision    0     Size range errors        0
802 chip restarts           0     Frame losses             0
Heartbeat losses            0
```

This command displays statistics about data transmitted across the link. All field values, except for those under Receive Filter, are summations. Over time, the values in these fields reach their maximum possible value. When this occurs, these fields can only be reset manually.

Transmits no error — The Transmits no error field specifies the number of frames that were successfully transmitted onto the medium. This includes the number of frames that were successfully transmitted on the first attempt, as well as frames that were successfully

transmitted after being deferred or that experienced one or more collisions. The maximum value of this 32-bit field is 4294967295.

Receives no error — This field specifies the number of frames that were successfully received over the medium. This includes the number of frames that were successfully received on the first attempt, as well as frames that were successfully received after being deferred or that experienced one or more collisions. The maximum value of this 32-bit field is 4294967295.

Transmits error — The Transmits error field specifies the number of transmission errors sent across the link. The value in this field specifies the number of frames which, due to errors on the link, were never transmitted. Although this value is commonly greater than 5% of the total number of frames transmitted, if it reaches or exceeds 5% of the total number of frames, check the hardware or check to see if the LAN is overloaded.

Receives error — The Receives error field specifies the number of transmission errors that were received from the link. The value in this field specifies the number of frames which were received, but were corrupted due to errors on the link. This value includes all frames which were discarded because of the setting of the current receive filter.

Out of Tx bufs (CIO card only) — The Out of Tx bufs field specifies the number of times that the LAN device adaptor (DA) reported to the driver that no transmit buffers were available. The maximum value of this 32-bit field is 4294967295.

Out of Rx bufs (CIO card only) — The Out of Rx bufs field indicates the number of times the LAN card reported to the driver that no receive buffers were available. This indicates only that the next buffer space was full and that the buffer pointer could not be incremented to an available buffer (the buffer pointer is incremented only after the driver requests the next frame). This also does not indicate that any frames were lost, however if another frame arrives before a receive buffer is made available, that frame will be lost. The value of this field should be very low. Retransmissions will occur if the link is out of Rx bufs. The maximum value for this 32-bit field is 4294967295.

Transmits deferred — The Transmits deferred field indicates the number of frames that deferred to other traffic before being transmitted onto the network. This means that the LAN card had to wait for carrier to drop, and stay dropped for 9.6 nanoseconds, before attempting to transmit the frame. This statistic only counts the number of frames that were deferred and later transmitted without collision.

Carrier losses — The Carrier losses field indicates that the transmitting node turned off the carrier signal on the cable. This occurred for one of the following reasons:

- The stub cable is not connected to the frontplane connector.
- The AUI (or AUI pigtail for ThinMAU) is not connected to the stub cable.
- The MAU is broken.
- If using thick LAN cable, there may be a short close to the MAU (ThinLAN cable shorts show up as a retry error as described in the Transmits 16 collision field description).

If the LAN continuously loses carrier, the problem is probably caused by a disconnected AUI or stub cable. Make sure that all connectors from the frontplane of the LAN hardware card to the MAU are connected securely.

NOTE

Collisions occur on IEEE 802.3 Local Area Network (LAN) links whenever two nodes on the link attempt to transmit data at the same time. When a collision occurs, the nodes which were involved in the collision each wait a random amount of time, called random backoff, before attempting to again transmit the packet along the link. If collisions continuously occur, check the terminators. Many of the fields described in this section are incremented whenever a collision occurs.

Transmits 1 retry — This field indicates the number of frames that collided once before being transmitted successfully. This means that the random backoff strategy was only used once.

Reflectometer (CIO card only) — The reflectometer field is similar in function to a TDR (Time Domain Reflectometer). The statistic holds the time count between the pulse and a reflection. Whenever a retry error occurs, the time in bit times (100ns) from when the frame started to transmit until the collision occurred is stored by this statistic. This can be useful for grossly determining the location of an opening in a cable, or possibly, a short in a ThinLAN cable. This field is erased after every transmit and is not updated after an external loopback frame is transmitted onto the link.

While this statistic may aid in pinpointing a problem without the need to do an actual TDR test, it should be noted that this statistic calculates the distance using a rough estimate (bit time) and can be inaccurate. This statistic should never be used as the only means of locating a cable fault. However, if this field is not equal to 0, then the hardware of the node is a likely cause of the failure.

The reflectometer field, for a thick LAN cable, is calculated in the following manner:

```
The ThickLAN velocity of propagation   =   .77c
Where c (the speed of light)           =   3x10 E8
The bandwidth of a LAN                 =   10Mb/sec.
```

Before determining the level of cable fault isolation, you must first determine how many meters of the cable are covered per bit time. You then divide .77c by 10Mb/sec. This translates into:

```
7x10E-2) x (3x10E8)
_________________   = 231 meters
10E6 b/sec)
```

Therefore, in order to pinpoint a fault in a thick LAN cable by the value of this field, multiply the field value by 231 meters.

The accuracy of the reflectometer field is plus or minus 1/2 bit time, or 115m. Using this calculation, the location of the cable fault is determined by the following formula:

```
(value of field x 231 meters) +\- 115m
```

Since the maximum length of a cable is 500m, the value of this field would be 0, 1 or 2, and would pinpoint a cable fault to 1 of 3 sections of cable.

If this value were to be used for isolating a cable fault in a ThinLAN cable, the value 0.65 would be substituted for .77c in the calculation above. (The ThinLAN velocity of propagation is .65c).

Transmits >1 retry — The Transmits >1 retry field indicates the number of frames that collided more than one, but fewer than 16 times, before being transmitted successfully onto the link. If the frame was not transmitted successfully (more than 16 attempts were made without success), then the card aborts transmission of this frame, and it counts the event as a retry error (see the Transmits 16 collision field).

CRC errors — The CRC errors field specifies the number of cyclic redundancy check (CRC) errors that were seen on the link. A CRC error indicates that the frame was checked using CRC-32 frame-checking, but that the value obtained by the CRC did not match the CRC value contained within the packet.

Normally there will be an equal number of alignment errors. If alignment errors occur frequently, one of the following may be the cause:

- A LAN card is not listening to the link before transmitting.
- A repeater that is performing poorly.
- A section of LAN coax which contains an impedance.
- The driver level of a MAU is set too low.

Transmits 16 collisions — The Transmits 16 collisions field indicates the number of times a frame or frames were not transmitted because 16 consecutive collisions occurred. This commonly occurs in the protocol during periods of high network utilization. If your node is experiencing continuous retry errors, the problem is most likely that a terminator

has been removed from the cable. Other possible causes include the following:

- There is an opening in the cable.
- If ThinLAN cable is used, the AUI may be disconnected.
- The LAN cable may shorted.

Whole byte errors — The Whole byte errors field is the number of frames received that were not an integer multiple number of bytes long. This occurs when an entire byte is not transmitted. This usually also causes the CRC error to be set.

Transmits late collision — This field indicates that a frame was active in the network for a longer time than is permitted by the protocol. The IEEE 802.3 protocol expects each frame to be transmitted within one slot time (the expected time for a 512 bit packet to traverse the entire network). The slot time exceeds the amount of time a single frame should need to traverse the entire network.

A value in this field indicates that a network problem caused a late collision. A late collision is one in which the collision occurs after one slot time has passed and another node, sensing that the network is inactive, begins to transmit a frame. Late collisions are caused by one of the following:

- Broken LAN cards in the network.
- A network that is too long.

A network can be made too long by installing too many repeaters between nodes. HP MAUs inform the LAN card of collisions after the 512 bit timer expires even though IEEE 802.3 standards do not require the MAU to monitor the link beyond that time. No attempt is made to retransmit a frame after a late collision.

Size range errors — The Size range errors field indicates the number of frames received that are not within the allowable size range. The allowable size range is 64–1518 bytes long. Unless the save bad frames bit is set on the LAN hardware card, the LAN hardware card throws these packets out.

802 chip restarts — The 802 chip restarts field was initially used to count the number of times that a specific version of the LAN chip locked up. This problem was remedied by a new version of that chip, however, this field still returns a value when one of the following events occurs:

- An AUI cable that is shorted and sending an intermittent signal to any of the connectors.
- Infinite deferral errors.
- “Jabbering” MAU.

- Noise from another node.
- Bad chips.

The value of the 802 chip restarts field provides information about the performance of the LAN card and the status of the LANCE chip status for overflow/underflow errors (this is monitored by firmware).

Frame losses — The frame losses field indicates the number of times that the LAN controller chip indicated that it has lost a frame. After some delta period of time following a transmission, no collision detect is seen. This is typically because there are no free receive buffers when a frame arrives.

Receives Dropped (NIO card only) — The Receives Dropped field indicates the total number of frames that were dropped because there was no receive buffer posted.

Receives Broadcast (NIO card only) — The Receives Broadcast field indicates the total number of frames received that were addressed to a broadcast address. If no broadcasts have been received, check the current receive filter to ensure that broadcasts are enabled. If broadcasts are enabled and no broadcasts have been received, this may be an indication of a faulty LAN card.

Receives Multicast (NIO card only) — The Receives Multicast field indicates the total number of frames received that were addressed to a multicast address. If no multicast frames are being received, check to make sure that the desired multicast address(es) are listed as part of the current multicast addresses.

Heartbeat losses — The Heartbeat losses field indicates that no SQE heartbeat was seen after a transmission and when IEEE 802.3 stub cable was connected. After a successful transmission, the 802.3 MAU sends an SQE message, called a “heartbeat,” through the Control In wire of the AUI. This heartbeat function lets the card know that the MAU is still functioning properly.

NOTE This statistic is not to be set if the Ethernet stub cable is connected.

NS 3000/iX IEEE 802.5/Token Ring Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX IEEE 802.5 Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   TR1   Linktype:   IEEE8025   Linkstate:   CONNECTED
```

Linkname — The Linkname field specifies the name of the link.

Linktype — The Linktype field specifies the type of link, such as LAP-B or IEEE 802.5, that is being monitored.

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Connected
- Not connected
- Retry

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for IEEE 802.5 links displays several fields in addition to the *LINKSTATE* parameter field. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=CONFIGURATION command:

```
Physical Path:     48       Data Rate:  16 Mbps
Current Station Address:      10-00-90-90-C8-46
Default Station Address:      10-00-90-90-C8-46
Functional Address Mask:            00-00-00-00
```

Physical Path — The Physical Path field displays the current physical path for the Token Ring card as specified in the NMMGR configuration file.

Data Rate — The data rate that the card is configured to operate at. This is set to either 4 or 16 Mbps by a jumper on the card.

Current Station Address — The Current Station Address field is a display of the six (6) byte address to which the node is configured to respond. This address is used whenever frames are sent to the network media. The default station address is used unless it is overridden in the NMMGR link configuration screen. If this field is changed, then the

station address of this node is changed. Make sure that you note this new address in the system manager log.

Default Station Address — The Default Station Address field is the default value for the Current Station Address described above. The default station address is determined by the specific Token Ring card.

Functional Address Mask — Bits set in this 4 octet field indicate functional addresses to which the Token Ring card may respond.

STATISTICS Parameter Fields

The *STATISTICS* parameter for IEEE 802.5 links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=STATISTICS command:

```
Physical Path:  48      Data Rate: 16 Mbps

Transmits no error:             0    Receives no error:             0
Transmit byte count:            0    Receive byte count:            0
Transmit errors:                0    Receive errors:                0
```

This command displays statistics about data transmitted across the link. All field values are summations. Over time, the values in these fields reach their maximum possible value. When this occurs, these fields can only be reset manually.

Transmits no error — The Transmits no error field specifies the number of frames that were successfully transmitted onto the medium. The maximum value of this 32-bit field is 4294967295.

Receives no error — This field specifies the number of frames that were successfully received over the medium. The maximum value of this 32-bit field is 4294967295.

Transmit byte count — This field specifies the transmit byte count.

Receive byte count — This field specifies the receive byte count.

Transmit errors — The Transmit errors field specifies the number of transmission errors sent across the link. The value in this field specifies the number of frames which, due to errors on the link, were never transmitted.

Receive errors — The Receive errors field specifies the number of transmission errors that were received from the link. The value in this field specifies the number of frames which were received, but were corrupted due to errors on the link. This value includes all frames which were discarded because of the setting of the current receive filter.

NS 3000/iX FDDI Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX FDDI Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   FDDILINK   Linktype:   FDDI   Linkstate:   RING UP
```

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for FDDI links displays several fields in addition to the LINKSTATE parameter field. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=CONFIGURATION command:

```
Physical Path:                0/28
Current Station Address:      08-00-09-09-63-67
Default Station Address:      08-00-09-09-63-67
Current Multicast Address List is empty
```

Physical Path — The Physical Path field displays the current physical path for the FDDI card as specified in the NMMGR configuration file.

Current Station Address — The Current Station Address field is a display of the six (6) byte address to which the node is configured to respond. This address is used whenever frames are sent to the network media. The default station address is used unless it is overridden in the NMMGR link configuration screen. If this field is changed, then the station address of this node is changed. Make sure that you note this new address in the system manager log.

Default Station Address — The Default Station Address field is the default value for the Current Station Address described above. The default station address is determined by the specific FDDI card.

Current Multicast Address List — This is a list of all multicast addresses that have been configured for this link. These multicast addresses are those to which this node will now respond.

STATISTICS Parameter Fields

The *STATISTICS* parameter for FDDI links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=STATISTICS command:

```
Transmits no error        41337  Receives no error        41859
SMT uni xmits             16347  SMT uni rec              16347
SMT non-uni xmits         16406  SMT non-uni rec          16347
SMT xmit bytes          2392533  SMT rec bytes          2386662
Transmit errors               0  SMT rec errors               0
Receive errors                4  Receives lost               26
Receives not copied           0  Receives discarded         212
```

This command displays statistics about the data transmitted across the link. All field values are summations.

Transmits no error — The number of frames that were successfully transmitted by this station. This count excludes all MAC frames.

Receives no error — The number of frames addressed to and successfully received by this station.

SMT uni xmits — The number of unicast frames transmitted by the SMT module on this station (reference: FDDI Station Management Standards document)

SMT uni rec — The number of unicast frames received by the SMT module on this station (reference: FDDI Station Management Standards document)

SMT non-uni xmits — The number of non-unicast (multicast and broadcast) frames transmitted by the SMT module on this station (reference: FDDI Station Management Standards document)

SMT non-uni rec — The number of non-unicast (multicast and broadcast) frames received by the SMT module on this station (reference: FDDI Station Management Standards document)

SMT xmit bytes — The number of bytes of data transmitted by the SMT module on this station.

SMT rec bytes — The number of bytes of data received by the SMT module on this station.

Transmit errors — The number of frames aborted or not transmitted by this station.

SMT rec errors — The number of frames with errors received by the SMT module on this station.

Receive errors — The number of all error frames that were detected by this station and no previous station.

Receives lost — The number of frames received that had an error.

Receives not copied — The number of frames addressed to this station but not copied into a receive buffer because there was no room for them.

Receives discarded — The number of frames received by this station that were discarded due to errors, invalid addresses, or lack of resources.

ALL Parameter Fields

The *ALL* parameter for FDDI links displays a combination of all fields presented by the LINKSTATE, CONFIGURATION, and STATISTICS commands. Please reference the previous sections for descriptions of the various fields.

DIAGNOSTIC Parameter Fields

The *DIAGNOSTIC* parameter for FDDI links displays all fields presented by the ALL parameter plus several additional fields that might be useful for HP representatives trying to debug FDDI link related problems. Brief descriptions will be given here; however, please note that some descriptions will be meaningful only to HP factory representatives. The following is an example of the additional fields displayed by this command.

```
Writes completed              0  Reads completed              0
Write bytes                   0  Read bytes                   0
Unicast writes                0  Unicast reads                0
Multicast writes              0  Multicast reads              0
Broadcast writes              0  Broadcast reads              0
Writes aborted                0  Reads aborted                0
Outbound high water           0  Non-routable reads           0
Num of heartbeats          3494  Subset buffer reqs           1
Number of users               1  Num of subset bufs          62
Num of power fails            0  Queued buffer reqs           0
Ring up time               3494  Read buffs in pool          63
Ring drops                    0  Read buffs avail            49
Loquix reinits                0  Read buffer size          5312
                                 SMT events                   8
Lan_in_active                ON
Configured                   ON
Ring_op                      ON

Reserved 1                    0  Reserved 2                   0
Reserved 3                    0  Reserved 4                   0
```

Writes completed — The number of data packets successfully sent from the driver to the FDDI card. The maximum value is two billion (2^31).

Reads completed — The number of packets received from the FDDI card by the driver. The maximum value is two billion (2^31).

Write bytes — The total number of bytes in all packets transmitted by the driver. The maximum value is one quadrillion (10^15).

Read bytes — The total number of bytes in all packets received by the driver. The maximum value is one quadrillion (10^15).

Unicast writes — The number of unicast packets transmitted by the driver.

Unicast reads — The number of unicast packets received by the driver.

Multicast writes — The number of multicast packets transmitted by the driver.

Multicast reads — The number of multicast packets received by the driver.

Broadcast writes — The number of broadcast packets transmitted by the driver.

Broadcast reads — The number of broadcast packets received by the driver.

Writes aborted — The number of times the transmit abort bit was set in the IO_RX_STATUS register at the completion of a packet transmission.

Reads aborted — The number of times the status length field in the read buffer trailer indicated that the received buffer is not valid.

Outbound high water — The maximum number of outbound packets queued within the driver awaiting transmission.

Non-routable reads — The number of inbound packets that did not have a valid destination SAP address.

Num of heartbeats — The number of heartbeat requests passed between the driver and the card. Also the number of seconds since the driver was configured (one heartbeat per second).

Subset buffer reqs — The number of times the driver has made a subset allocation buffer manager request.

Number of users — The number of network transports that have configured with the driver.

Num of subset bufs — The total number of buffers received from all the subset allocation requests made during the life of the driver. Dividing this number by the Subset buffer reqs value yields the average number of buffers returned per request.

Num of power fails — The number of times that a system powerfail has been detected during the life of the driver.

Queued buffer reqs — The number of times the driver has made a queued buffer manager request.

Ring up time — The number of seconds the FDDI ring has been up since the driver was started. Subtracting this number from the Num of heartbeats value will provide the number of seconds that the ring has been down. Note the ring being down does not by itself indicate a problem with this node. The ring will not be up unless the FDDI concentrator and other nodes are configured and active.

Read buffs in pool — The number of buffers in the inbound buffer pool.

Ring drops — The number of times the ring down signal has been received during the life of the driver.

Read buffs avail — The number of buffers in the inbound buffer pool that are currently queued within the driver.

Loquix reinits — The number of times the Loquix chip has been reinitialized.

Read buffer size — The size in bytes of the buffers in the inbound buffer pool.

SMT events — The number of SMT events received by the driver.

Flag Status

The status of several flags are printed here. If a flag's status is not reported here then that flag is NOT set. The meaning each flag is briefly described here.

Io_tx — The IO_TX_STATUS register has been read since the last card interrupt.

Io_cmd — The IO_CMD_STATUS register has been read since the last card interrupt.

Lan_in_active — An inbound DMA buffer has been posted to the card.

Lan_out_active — An outbound DMA is currently active.

Ctrl_out_active — An outbound control operation is currently active.

Configured — The FDDI card has been successfully configured.

Trace_on — Tracing has been enabled for this FDDI link.

Reset_on — The FDDI card is currently being reset.

Pfail_on — The FDDI card is currently recovering from a powerfail.

Bmgr_queued_aloc_on — A buffer manager queued allocation request is pending.

Config_on — The FDDI card is currently being configured.

Post_read_pending — The driver is temporarily out of inbound buffers to post to the card.

Ring_op — The FDDI card is signaling that the FDDI ring is operational.

Ctrl_response_pending — The driver has a control response pending on the card.

Free_space_pending — A free space request is pending against the card.

Bad_card_on — A problem with the FDDI card has been detected.

Do_bind_on — The FDDI driver is being started and initialized.

Download_on — The FDDI card firmware is being downloaded to the card.

Statistics_on. A statistics request is pending against the card.

Reserved 1: Reserved for future use.

Reserved 2: Reserved for future use.

Reserved 3: Reserved for future use.

Reserved 4: Reserved for future use.

NS 3000/iX 100VG-AnyLAN Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX 100VG-AnyLAN Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   SYSLINK   Linktype:   VG8023   Linkstate:   CONNECTED
```

Linkname — The Likname field specifies the name of the link.

Linktype — The Linktype field specifies the type of link, such as LAP-B, 100VG-802.3, or IEEE 802.3, that is being monitored.

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Connected
- Not connected

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for 100VG-AnyLAN links displays several fields in addition to the LINKSTATE parameter field. This is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=CONFIGURATION command:

```
Physical Path:                10/4/8
Current Station Address:      08-00-09-DD-CC-99
Default Station Address:      08-00-09-DD-CC-99
Current Multicast Addresses:
  99-00-09-00-00-01  09-00-09-00-00-03  09-00-09-00-00-04  09-00-09-00-00-06
```

Physical Path — The Physical Path field displays the current physical path for the adapter card as specified in the NMCONFIG configuration file.

Current Station Address — The Current Station Address field is a display of the six (6) byte address to which the node is configured to respond. This address is used whenever frames are sent to the network media. The default station address is used unless it is overridden in the NMMGR link configuration screen. If this field is changed, then the station address of this node is changed. Make sure that you note this new address in the system manager log.

Default Station Address — The Default Station Address field is the default value for the Current Station Address described previously. The

default station address is determined by the specific adapter card. It is also printed on a small label attached to a circuit board on the adapter card. If the adapter card is changed for any reason, the Default Station Address of this node will change.

Current Multicast Address List — The Current Multicast Addresses field contains a list of all multicast addresses to which the adapter card responds. The default multicast address list contains no addresses. If no multicast addresses are enabled, the follow message is printed:

```
Current multicast address list is empty
```

Multicast addresses are configured automatically by the network transport(s) using the adapter card. The maximum number of multicast addresses allowed is 16. An example of multicast addresses are:

`09-00-09-00-00-01`	Probe address
`09-00-09-00-00-02`	Second probe address
`09-00-09-00-00-04`	DTC boot address

STATISTICS Parameter Fields

The *STATISTICS* parameter for 100VG-AnyLAN links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. For an example of the data that is displayed when you issue the `LINKCONTROL` *linkname*`;STATUS=STATISTICS` command:

```
Transmit bytes norm      1456044817     Receive bytes norm        5308537515
Transmit byets hipri          62029     Receive bytes hipri              0
Transmits norm              2707747     Receives norm           5302073790
Transmits hipri                712     Receives hipri                   0
Transmits no error          2708459     Recv broadcast norm        47068412
Transmits dropped                 0     Recv broadcast hipri             0
Trans underruns                   0     Recv multicast norm        1300291
Recv overruns                     0     Recv multicast hipri             0
Recv deferred                   123     Receives no error        664709473
CRC or Maxsize error              0     Recv dropped: addr        16002992
Code or Align error               0     Recv dropped: buffer           262
Link disconnects                  0     Recv driooedL dna                0
Link speed                      100     Recv dropped: other          24785
Link mode                     100VG     Secs since clear           6173798
Link training result      CONNECTED
```

NOTE Some of the parameter descriptions vary according to whether the adapter card is operating at 100Mbps or 10Mbps speed. For a 100VG adapter card operating in 10Base-T mode, refer to 100Base-T statistics.

This command displays statistics about data transmitted and received across the link. Many field values are summations. Over time, the

values in these fields reach their maximum possible value. When this occurs, these fields can only be reset manually.

Transmit bytes norm — Total number of bytes successfully transmitted onto the medium at normal priority. This includes unicast, broadcast, and multicast frames. It also includes frames for which normal priority was requested, but which were later automatically boosted to demand priority by the 100VG-AnyLAN network. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the example above, 1.45 billion bytes were transmitted, or about 1.38 Gbytes.

Transmit bytes hipri — Total number of bytes successfully transmitted onto the medium at high (demand) priority. This includes unicast, broadcast, and multicast frames. It does not include frames for which normal priority was originally requested, but which were later automatically priority-boosted by the 100VG-AnyLAN network. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the example above, 62,029 high priority bytes have been transmitted, or about 60K bytes.

Transmits norm — Total number of frames successfully transmitted onto the medium at normal priority. This includes unicast, broadcast, and multicast frames. It also includes frames for which normal priority was requested, but which were later automatically boosted to demand priority by the 100VG-AnyLAN network. The byte count given by Transmit bytes norm is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 2.71million frames were transmitted at normal priority.

Transmits hipri — Total number of frames successfully transmitted onto the medium at high (demand) priority. This includes unicast, broadcast, and multicast frames. It does not include frames for which normal priority was originally requested, but which were later automatically priority-boosted by the 100VG network. The byte count given by Transmit bytes hipri is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the example above, 712 frames were transmitted at high priority.

Transmits no error — Total number of frames the adapter card reports it successfully transmitted onto the medium. These adapter card statistics are periodically read, and are accumulated by the link driver. The total includes all unicast, broadcast, and multicast frames, at both normal and high (demand) priority. It should equal the sum of (Transmits norm + Transmits hipri). The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 2.71 million frames were transmitted regardless of priority level.

Transmits dropped — Total number of frames the link driver discarded because the transmit queue was full, or because the data to be sent was fragmented beyond recognition. The maximum value of this 32-bit field is 2147483647. It would be unusual for this statistic to contain a nonzero value.

Trans underruns — Total number of frames aborted by the adapter card during transmission because the remaining data was not made available to the transmit hardware fast enough. It indicates unexpected latency on the dedicated internal bus onboard the adapter card. If this condition occurs, the adapter will automatically adjust to improve the latency, and retransmit the aborted frame automatically. The maximum value of this 32-bit field is 2147483647. For this statistic, a value of less than 3 would be considered normal.

Recv overruns — Total number of address-matched frames that could not be received into the adapter card, either because prior data was not being removed by the receive hardware fast enough, or because their size exceeded the maximum frame size. May indicate unexpected latency on the dedicated internal bus onboard the adapter card, which cannot be automatically adjusted by the link driver. The maximum value of this 32-bit field is 2147483647. It would be unusual for this statistic to contain a nonzero value.

Recv deferred — Number of times an address-matched receive frame was temporarily held (queued) by the link driver, because of a momentary lack of DMA resources. Once those resources became available, the frame was automatically transferred to the host. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 123 frames were temporarily held.

CRC or Maxsize error — Number of cyclic redundancy check (CRC) errors or oversized frames that were seen during reception by the link. A CRC error indicates that the frame was checked using CRC-32 frame-checking, but that the value obtained by the CRC did not match the CRC value contained at the end of the frame. CRC errors do not include frames having alignment or coding errors. Oversized frames are those longer than 1518 bytes. These adapter card statistics are periodically read, and are accumulated by the link driver. The maximum value of this 32-bit field is 2147483647. A nonzero value in this statistic could indicate a defective cable, adapter, or hub, a loose connection, presence of severe electrical noise along the cable path, or a misbehaving application, adapter, or hub at the transmission end.

Code or Align error — Number of frames received with an alignment error (not an even multiple of 4 bits of data) or code errors (an error signal was received from the 100VG-AnyLAN receive hardware). These adapter card statistics are periodically read, and are accumulated by the link driver. The maximum value of this 32-bit field is 2147483647. If

alignment errors occur frequently, one of the following may be the cause:

- A 100VG-AnyLAN adapter card is not operating to within 802.12 specifications.
- A 100VG-AnyLAN hub is performing poorly.
- The 100VG-AnyLAN cable is not CAT-3 or CAT-5 grade.
- A section of 100VG-AnyLAN cable contains wire pairs which are not properly twisted, paired, or of equal length.

Link disconnects — Number of times the link driver noticed the link had previously been established, but was no longer up. This may occur because the cable was unplugged, the hub was powered off, the hub automatically requested a reconnect, or normally (at link shutdown time). This total does not include repetitive, failed attempts by the link driver to reestablish the link. The maximum value of this 32-bit field is 2147483647.

Link speed — Maximum link speed (either 100 or 10) in millions of bits per second, the link is currently configured to operate at. When displaying a 100VG-AnyLAN link operating in 100VG-AnyLAN mode, this value will always be 100. When a 100VG-AnyLAN link is operating in 10Base-T mode, this value will always be 10.

Link mode — Electrical mode the link is currently operating at. When displaying a 100VG-AnyLAN link operating in 100VG-AnyLAN mode, this value will always be “100VG-AnyLAN”. When a 100VG-AnyLAN link is operating in 10Base-T mode, this value will be “10Base-T”.

Link training result — The result of the last automatic 100VG-AnyLAN “link training” operation performed during link establishment. If the cable is connected to an operational 100VG-AnyLAN hub port, the link is configured for 100VG-AnyLAN operation in NMCONFIG, and the link driver has been started, this value should be “CONNECTED”. Most other values indicate a problem with the hub port, the connection to that 100VG-AnyLAN hub port, or the cable grade is not CAT-3 or CAT-5.

Receive bytes norm — Total number of bytes successfully received over the medium at normal priority. This includes unicast, multicast, broadcast, and link training frames. It does not include frames received by the adapter card but dropped because no upper layer protocol had requested those frames, or because the link was disconnected. The maximum value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the example above, 5.30 billion bytes were received, or about 5.06 Gbytes.

Receive bytes hipri — Total number of bytes successfully received over the medium at high (demand) priority. This includes unicast, multicast, and broadcast frames. It also includes frames for which

normal priority was originally requested by the sender, but which were later automatically priority-boosted by the 100VG-AnyLAN network. It does not include frames received by the adapter card but dropped because no upper layer protocol had requested those frames, or because the link was disconnected. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the example above, no high priority data has been received.

Receives norm — Total number of unicast frames (addressed to this specific adapter card) which were successfully received over the medium at normal priority and forwarded to an upper layer protocol such as IP. This includes link training frames received from the 100VG-AnyLAN hub. It does not include unicast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 2.07 million unicast frames were received at normal priority and forwarded to upper layers.

Receives hipri — Total number of unicast frames (addressed to this specific adapter card) which were successfully received over the medium at high (demand) priority and forwarded to an upper layer protocol such as IP. This includes unicast frames for which normal priority was originally requested by the sender, but which were later automatically priority-boosted by the 100VG-AnyLAN network. It does not include unicast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, no frames have been received at high priority.

Recv broadcast norm — Total number of frames addressed to a broadcast address which were successfully received over the medium at normal priority and forwarded to an upper layer protocol such as ARP. This does not include broadcast frames received but dropped for any reason. The maximum value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 47.1 million broadcast frames were received at normal priority and forwarded to upper layers.

Recv broadcast hipri — Total number of frames addressed to a broadcast address which were successfully received over the medium at high (demand) priority and forwarded to an upper layer protocol such as ARP. This includes broadcast frames for which normal priority was originally requested by the sender, but which were later automatically priority-boosted by the 100VG-AnyLAN network. It does not include broadcast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, no frames have been received at high priority.

Recv multicast norm — Total number of frames addressed to a multicast address which were successfully received over the medium at normal priority and forwarded to an upper layer protocol such as

PROBE. This does not include multicast frames received but dropped for any reason. Upper layer protocols register desired multicast addresses with the link driver during initialization. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 1.30 million multicast frames were received at normal priority and forwarded to upper layers.

Recv multicast hipri — Total number of frames addressed to a multicast address which were successfully received over the medium at high (demand) priority and forwarded to an upper layer protocol such as PROBE. This includes multicast frames for which normal priority was originally requested by the sender, but which were later automatically priority-boosted by the 100VG-AnyLAN network. It does not include multicast frames received but dropped for any reason. Upper layer protocols register desired multicast addresses with the link driver during initialization. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, no frames have been received at high priority.

Receives no error — Total number of frames the adapter card reports it successfully received over the medium. These adapter card statistics are periodically read, and are accumulated by the link driver. This includes both normal and high (demand) priority frames. It does not include any frames the adapter card detected errors against. The total should approximate the sum of all frames forwarded to upper layer protocols, plus all frames dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 6.64 million frames were received successfully by the adapter card, then either forwarded or dropped.

Recv dropped: addr — Total number of frames received by the adapter card, but dropped because no upper layer protocol had requested future reception of those frames, or because that protocol unbound itself from the link while the received frame was still in motion. Older, intelligent adapter cards can invisibly receive and drop these frames, often without ever reporting them as statistics. But today's adapters are not intelligent, and require link driver involvement. For users unfamiliar with seeing it, this statistic may seem excessive. But it is important because it gives an indication of the amount of unnecessary traffic present on the network segment to which the adapter card is connected. High values may indicate a need to resegment the network, since systems and their adapters are spending a large amount of time and resources recognizing and dropping frames they do not care to see. High volumes of such traffic can also limit network bandwidth. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, out of 66.4 million frames received, 16.0 million have been dropped based on address: about 24% of all frames received are then being dropped. This level would be considered typical or perhaps a little high.

Recv dropped: buffer — Total number of frames received from the adapter card, but dropped because no data buffers were available from the upper layer protocol requesting to receive these frames. This is a relatively common occurrence, amounting to flow control for all protocols sharing those buffers. Many protocols include built-in mechanisms for detecting lost frames and requesting their retransmission from the remote side. The maximum value of this 32-bit field is 2147483647. In the previous example, out of 66.4 million frames received, 262 have been dropped for lack of buffer resources; any similar level would be considered normal.

Recv dropped: dma — The link driver design now queues frames under conditions of low DMA resources, so this statistic is now obsolete, should never contain a nonzero value, and may be deleted in a future release.

Recv dropped: other — Sum total number of frames received from the adapter card, but dropped because: an upper layer protocol error was returned; the required address format was not supported; the frame arrived while the link driver was in an unusual state; or for perfect multicast filtering reasons. The maximum value of this 32-bit sum is 2147483647. In the previous example, 24785 frames have been dropped for one or more of the reasons listed; this number would be considered high, and further investigation might be needed if it appears to be impacting any applications.

Secs since clear — The number of seconds elapsed since statistics were last reset via the `LINKCONTROL` *`linkname`*`; STATUS=RESET` command. This gives the sample time over which the displayed statistics have been collected. Per-time-unit figures may then be calculated if desired. The maximum value of this 32-bit field is 2147483647, or about 68 years.

NS 3000/iX 100Base-T Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX 100Base-T Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:   SYSLINK   Linktype:   PCI 100BT  Linkstate:   CONNECTED
```

Linkname — The Likname field specifies the name of the link.

Linktype — The Linktype field specifies the type of link, such as LAP-B, 100BT, or IEEE 802.3, that is being monitored, and the type of I/O bus it is installed on (will be preceded with "PCI" for PCI 100Base-T links).

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Connected
- Not connected

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for 100Base-T links displays several fields in addition to the LINKSTATE parameter field. This is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=CONFIGURATION command:

```
Physical Path:                10/4/8
Current Station Address:      08-00-09-DD-A2-5C
Default Station Address:      08-00-09-DD-A2-5C
Current Multicast Addresses:
  99-00-09-00-00-01  09-00-09-00-00-03  09-00-09-00-00-04  09-00-09-00-00-06
```

Physical Path — The Physical Path field displays the current physical path for the adapter card as specified in the NMCONFIG configuration file.

NOTE On PCI 100Base-T links, the physical path generally has more components and is a "longer" path (e.g., 1/12/0/0). Otherwise, output for PCI links is the same as HP-PB.

Current Station Address — The Current Station Address field is a display of the six (6) byte address to which the node is configured to respond. This address is used whenever frames are sent to the network

media. The default station address is used unless it is overridden in the NMMGR link configuration screen. If this field is changed, then the station address of this node is changed. Make sure that you note this new address in the system manager log.

Default Station Address — The Default Station Address field is the default value for the Current Station Address described above. The default station address is determined by the specific adapter card. On HP-PB systems, it is also printed on a small label attached to a circuit board on the adapter card.

NOTE This printed label is not available for PCI adapter cards. If the adapter card is changed for any reason, the Default Station Address of this node will change.

Current Multicast Address List — The Current Multicast Addresses field contains a list of all multicast addresses to which the adapter card responds. The default multicast address list contains no addresses. If no multicast addresses are enabled, the follow message is printed:

```
Current multicast address list is empty
```

Multicast addresses are configured automatically by the network transport(s) using the adapter card. For PCI 100BT links, the maximum number of multicast addresses currently allowed is 14 (this is due to an implementation decision regarding how the PCI 100BT card manages multicast addresses). An example of multicast addresses are:

`09-00-09-00-00-01`	Probe address
`09-00-09-00-00-02`	Second probe address
`09-00-09-00-00-04`	DTC boot address

STATISTICS Parameter Fields

The *STATISTICS* parameter for 100Base-T links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. For an example of the data that is displayed when you issue the `LINKCONTROL` *`linkname`*`;STATUS=STATISTICS` command:

```
Transmit bytes              10191188     Receive bytes              495231926
Transmits                     127700     Receives                      113968
Transmits no error            127700     Receives broadcast           3955124
Transmits dropped                  0     Receives multicast           1743137
Transmits deferred              1738     Receives no error           15688503
Transmits 1 retry                 86     CRC or Maxsize error              0
Transmits >1 retry               179     Code or Align error               0
Trans 16 collisions                0     Recv dropped: addr          9876255
Trans late collision               0     Recv dropped: buffer              0
Trans underruns                    0     Recv dropped: dma                 0
```

```
Carrier losses                    0     Recv dropped: other              19
Link disconnects                  0     Recv deferred                     0
Link speed                       10     Recv overruns                     0
Link duplex                    Half     Link auto sensed                 No
Link mode                 100Base-TX     Secs since clear            2602760
```

The following example of the LINKCONTROL output for a PCI 100Base-T link (some of the statistics differ for PCI).

```
Transmit bytes             10191188     Receive bytes             495231926
Transmits                    127700     Receives unicast             113968
Transmits no error           127700     Receives broadcast          3955124
Transmits dropped                 0     Receives multicast          1743137
Transmits deferred             1738     Receives no error          15688503
Transmits 1 retry                86     Recv CRC error                    0
Transmits >1 retry              179     Recv maxsize error                0
Trans 16 collisions               0     Recv dropped: addr          9876255
Trans late collision              0     Recv dropped: buffer              0
Trans underruns                   0     Recv dropped: descr               0
Carrier losses                    0     Recv dropped: other              19
Trans jabber timeout              0     Recv watchdg timeout              0
Link disconnects                  0     Recv collisions                   0
Link speed                       10     Recv overruns                     0
Link duplex                    Half     Link auto sensed                 No
Link mode          100Base-TX Addon     Secs since clear               5259
```

NOTE Some of the parameter descriptions vary according to whether the adapter card is operating at 100Mbps or 10Mbps speed. Most also apply to an HP-PB 100VG-AnyLAN card operating in 10Base-T mode.

This command displays statistics about data transmitted and received across the link. Many field values are summations. Over time, the values in these fields reach their maximum possible value. When this occurs, these fields can only be reset manually.

Transmit bytes — Total number of bytes successfully transmitted onto the medium. This includes unicast, broadcast, and multicast frames that were successfully transmitted on the first attempt, as well as frames that were successfully transmitted after being deferred or that experienced one or more collisions. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 10.1 million bytes were transmitted, or about 9.7 Mbytes.

Transmits — Total number of frames successfully transmitted onto the medium at normal priority. This includes unicast, broadcast, and multicast frames that were successfully transmitted on the first attempt, as well as frames that were successfully transmitted after being deferred or that experienced one or more collisions. The byte count given by Transmit bytes is distributed over this number of

frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 127700 frames were transmitted at normal priority.

Transmits no error — Total number of frames the adapter card reports it successfully transmitted onto the medium. These adapter card statistics are periodically read, and are accumulated by the link driver. The total includes all unicast, broadcast, and multicast frames that were successfully transmitted on the first attempt, as well as frames that were successfully transmitted after being deferred or that experienced one or more collisions. It should equal the value of "Transmits". The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 127700 frames were transmitted.

Transmits dropped — Total number of frames the link driver discarded because the transmit queue was full, or because the data to be sent was fragmented beyond recognition. The maximum value of this 32-bit field is 2147483647. It would be unusual for this statistic to contain a nonzero value.

NOTE

Collisions occur on 100Base-T Local Area Network links whenever two nodes on the link attempt to transmit data at the same time. When a collision occurs, each node involved in the collision waits a random amount of time, called random backoff, before attempting to again transmit the frame along the link. Many of the fields described in this section are incremented whenever a collision occurs.

Transmits deferred — Total number of frames that were deferred to other network traffic before their initial transmission onto the network. This means that the 100Base-T card had to wait for carrier to drop and stay dropped for a few nanoseconds, before attempting to transmit the frame. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 1738 transmit frames were deferred.

Transmits 1 retry — This field indicates the number of frames that collided once before being transmitted successfully. This means that the random backoff strategy was only used once. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the example above, 86 frames were transmitted after only one collision. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0.

Transmits >1 retry — This field indicates the number of frames that collided more than once, but fewer than 16 times, before being transmitted successfully onto the link. If the frame was not transmitted successfully (more than 16 attempts were made without success), then the card aborts transmission of this frame, and it counts the event in the "Trans 16 collisions" field. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the

previous example, 179 frames were transmitted after experiencing between 2 and 15 collisions. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0.

Trans 16 collisions — This field indicates the number of times a frame was not transmitted, because 16 consecutive collisions occurred. This can occur during periods of high network utilization. If the node is experiencing many Trans 16 errors, possible causes include the following:

- The network is saturated with traffic.
- There is a short in the cable.
- There is an opening in the cable.

The maximum value of this 32-bit field is 2147483647. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0.

Trans late collision — This field indicates that a frame was active in the network for a longer time than is permitted by the protocol. No attempt is made to retransmit a frame after a late collision. The IEEE 802.3 protocol expects each frame to be transmitted within one slot time (the expected time for a 512 bit frame to traverse the entire network). The slot time exceeds the amount of time a single frame should need to traverse the entire network.

A value in this field indicates that a network problem caused a late collision. A late collision is one in which the collision occurs after one slot time has passed and another node, sensing that the network is inactive, begins to transmit a frame. Late collisions are caused by one of the following:

- A network that is too long.
- Broken 100Base-T cards in the network.

A network can be made too long by installing too many repeaters between nodes. The 100Base-T card hardware detects collisions after the 512 bit timer expires even though IEEE 802.3 standards do not require the link to be monitored beyond that time. The maximum value of this 32-bit field is 2147483647. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0.

Trans underruns — Total number of frames aborted by the adapter card during transmission because the remaining data was not made available to the transmit hardware fast enough. It indicates unexpected latency on the dedicated internal bus onboard the adapter card. If this condition occurs, the adapter will automatically adjust to improve the latency, and retransmit the aborted frame automatically. The maximum value of this 32-bit field is 2147483647. For this statistic, a value of less than 3 would be considered normal.

Carrier losses — This field indicates that the transmitting node turned off the carrier signal on the cable. A carrier loss occurs when a receive carrier was not detected after a slottime from the start of transmission. The carrier must be present continuously from the start until the end of transmission to prevent an error. If the 100Base-T link continuously loses carrier, the problem is probably caused by a faulty hub or cable, or a disconnected cable somewhere else within the network, along the path between the target nodes. The maximum value of this 32-bit field is 2147483647. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0.

NOTE The PCI 100Base-T adapter card reports both “loss of carrier” (during transmission) and “no carrier” (no carrier detected at start of transmission) as distinct events. For PCI, the “carrier losses” statistic represents the sum of the “no carrier” and “loss of carrier” events.

Trans jabber timeouts — The number of times the adapter card transmitted onto the LAN for too long, and had to be forced to stop. This should only occur if the adapter card or hub are faulty. The maximum value of this 32-bit field is 2147483647. This statistic is displayed for PCI 100Base-T only.

Link disconnects — Number of times the link driver noticed the link had previously been established, but was no longer up. This may occur because the cable was unplugged, the hub was powered off, the hub automatically requested a reconnect, or normally (at link shutdown time). This total does not include repetitive, failed attempts by the link driver to reestablish the link. The maximum value of this 32-bit field is 2147483647.

Link speed — Maximum link speed (either 100 or 10) in million bits per second, the link is currently configured to operate at. When displaying a 100Base-T link operating in 100Base-T mode, this value will always be 100. When a 100Base-T link is operating in 10Base-T mode, this value will always be 10.

Link mode — Electrical mode the link is currently operating at. When displaying a 100Base-T link operating in 100Base-T mode, this value will always be “100Base-T”. When it is operating in 10Base-T mode, this value will be “10Base-T”. For PCI, the type of card (e.g., “Addon”) may also be shown here.

Receive bytes — Total number of bytes successfully received over the medium. This includes unicast, multicast, broadcast, and that were successfully received on the first attempt, as well as frames that were successfully received after being deferred or that experienced one or more collisions. The maximum value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 495 million bytes were received, or about 472 Mbytes.

Receives / Receives Unicast — Total number of unicast frames (addressed to this specific adapter card) which were successfully received over the medium and forwarded to an upper layer protocol such as IP. This includes unicast frames that were successfully received on the first attempt, as well as unicast frames that were successfully received after being deferred, or that experienced one or more collisions. It does not include unicast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 113968 unicast frames were received and forwarded to upper layers.

NOTE This has been renamed "Receives Unicast" in the PCI statistics display to better distinguish this statistic from the other receive statistics, but the meaning is the same as before.

Receives broadcast — Total number of frames addressed to a broadcast address which were successfully received over the medium and forwarded to an upper layer protocol such as IP. This includes broadcast frames that were successfully received on the first attempt, as well as broadcast frames that were successfully received after being deferred, or that experienced one or more collisions. It does not include broadcast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 3.96 million broadcast frames were received and forwarded to upper layers.

Receives multicast — Total number of frames addressed to a multicast address which were successfully received over the medium and forwarded to an upper layer protocol such as IP. This includes multicast frames that were successfully received on the first attempt, as well as multicast frames that were successfully received after being deferred, or that experienced one or more collisions. It does not include multicast frames received but dropped for any reason. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 1.74 million multicast frames were received and forwarded to upper layers.

Receives no error — Total number of frames the adapter card reports it successfully received over the medium. These adapter card statistics are periodically read, and are accumulated by the link driver. This includes all address-matched unicast, broadcast, and multicast frames received without error. It does not include frames not addressed to the adapter card, frames dropped for any reason, or any frames the card detected errors against. The total should approximate the sum of all frames forwarded to upper layer protocols. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the example above, 15.7 million frames were received successfully by the adapter card, then either forwarded or dropped.

CRC or Maxsize error — Number of cyclic redundancy check (CRC) errors or oversized frames that were seen during reception by the link. A CRC error indicates that the frame was checked using CRC-32 frame-checking, but that the value obtained by the CRC did not match the CRC value contained at the end of the frame. CRC errors do not include frames having alignment or coding errors. Oversized frames are those longer than 1518 bytes. These adapter card statistics are periodically read, and are accumulated by the link driver. The maximum value of this 32-bit field is 2147483647. A nonzero value in this statistic could indicate a defective cable, adapter, or hub, a loose connection, presence of severe electrical noise along the cable path, or a misbehaved application, adapter, or hub at the transmission end. For PCI 100Base-T, these statistics (CRC error, Maxsize error) are reported separately by the adapter card and are displayed separately (Recv CRC error, Recv Maxsize error).

Code or Align error — Number of frames received with an alignment error (not an even multiple of 4 bits of data) or code errors (an error signal was received from the 100Base-T receive hardware). These adapter card statistics are periodically read, and are accumulated by the link driver. The maximum value of this 32-bit field is 2147483647. This statistic is not applicable to PCI and is not displayed for PCI 100Base-T links. If alignment errors occur frequently, one of the following may be the cause:

- A 100Base-T adapter card is not operating to within 802.3 specifications.
- A 100Base-T hub is performing poorly.
- The 100Base-T cable is not CAT-5 grade.
- A section of 100Base-T cable contains wire pairs which are not properly twisted, paired, or of equal length.

Recv dropped: addr — Total number of frames received by the adapter card, but dropped because no upper layer protocol had requested future reception of those frames, or because that protocol unbound itself from the link while the received frame was still in motion. Older, intelligent adapter cards can invisibly receive and drop these frames, often without ever reporting them as statistics. But today's adapters are not intelligent, and require link driver involvement. For users unfamiliar with seeing it, this statistic may seem excessive. But it is important because it gives an indication of the amount of unnecessary traffic present on the network segment to which the adapter card is connected. High values may indicate a need to resegment the network, since systems and their adapters are spending a large amount of time and resources recognizing and dropping frames they do not care to see. High volumes of such traffic can also limit network bandwidth. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, out of 15.7 million frames received, 9.87 million have been dropped

based on address: about 63 percent of all frames received are then being dropped. This level would be considered high.

Recv dropped: buffer — Total number of frames received from the adapter card, but dropped because no data buffers were available from the upper layer protocol requesting to receive these frames. This is a relatively common occurrence, amounting to flow control for all protocols sharing those buffers. Many protocols include built-in mechanisms for detecting lost frames and requesting their retransmission from the remote side. The maximum value of this 32-bit field is 2147483647. In the previous example, out of 66.4 million frames received, 262 have been dropped for lack of buffer resources; any similar level would be considered normal.

Recv dropped: dma — The link driver design now queues frames under conditions of low DMA resources, so this statistic is now obsolete, should never contain a nonzero value, and may be deleted in a future release. This statistic is not applicable to PCI and is not displayed for PCI 100Base-T links.

Recv dropped descr — Total number of times the adapter card reported it was unable to receive a frame because of a problem with the receive instructions specified by the software driver. The maximum value of this 32-bit field is 2147483647. It would be unusual for this statistic to contain a nonzero value. This statistic is only applicable for PCI 100Base-T links.

Recv dropped: other — Sum total number of frames received from the adapter card, but dropped because: an upper layer protocol error was returned; the required address format was not supported; the frame arrived while the link driver was in an unusual state; or for perfect multicast filtering reasons. The maximum value of this 32-bit sum is 2147483647. In the previous example, 24785 frames have been dropped for one or more of the reasons listed; this number would be considered high, and further investigation might be needed if it appears to be impacting any applications.

Recv deferred — Number of times an address-matched receive frame was temporarily held (queued) by the link driver, because of a momentary lack of DMA resources. Once those resources became available, the frame was automatically transferred to the host. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 123 frames were temporarily held. This statistic is not applicable to PCI and is not displayed for PCI 100Base-T links.

Recv overruns — Total number of address-matched frames that could not be received into the adapter card, either because prior data was not being removed by the receive hardware fast enough, or because their size exceeded the maximum frame size. May indicate unexpected latency on the dedicated internal bus onboard the adapter card, which cannot be automatically adjusted by the link driver. The maximum

value of this 32-bit field is 2147483647. It would be unusual for this statistic to contain a nonzero value.

The following receive statistics are only applicable for PCI 100Base-T links:

Recv watchdg timeout — The number of times the adapter card tried to receive from the LAN for too long, and had to be forced to stop. This should only occur if the adapter card, the hub, or another adapter card on the network, are faulty. The maximum value of this 32-bit field is 2147483647.

Recv collisions — This field indicates that a frame was active in the network for a longer time than is permitted by the protocol, and damaged a frame being received. No attempt is made to re-receive a frame damaged by a late collision. The IEEE 802.3 protocol expects each frame to be transmitted within one slot time (the expected time for a 512 bit frame to traverse the entire network). The slot time exceeds the amount of time a single frame should need to traverse the entire network.

A value in this field indicates that a network problem caused a late collision. A late collision is one in which the collision occurs after one slot time has passed and another node, sensing that the network is inactive, begins to transmit a frame. Late collisions are caused by one of the following:

- A network that is too long.
- Broken 100Base-T cards in the network.

A network can be made too long by installing too many repeaters between nodes. The 100Base-T card hardware detects receive collisions occurring after the 512 bit timer expires even though IEEE 802.3 standards do not require the link to be monitored beyond that time. The maximum value of this 32-bit field is 2147483647. This statistic is not logged when the adapter card is operating in full duplex mode, and would print as 0. Not applicable to HP-PB.

Link auto sensed — A value of Yes indicates the local adapter card attempted to autonegotiate its speed and duplex settings, and the remote side returned a set of remote capabilities to the local adapter card, meaning the remote also supports the autonegotiation feature. The “Link speed” and “Link mode” fields report the settings chosen by the two sides. A value of No indicates the remote side did not return a set of capabilities although the local adapter card attempted to autonegotiate, and the “Link speed” and “Link mode” fields report the driver’s best-guess settings. “No” is also shown when the local adapter card is configured in NMCONFIG to use fixed speed and duplex settings.

Secs since clear — The number of seconds elapsed since statistics were last reset via the `LINKCONTROL linkname; STATUS=RESET` command. This gives the sample time over which the displayed statistics have been collected. Per-time-unit figures may then be calculated if desired. The maximum value of this 32-bit field is 2147483647, or about 68 years.

NS 3000/iX LAPBMUX Link Statistics

The following section describes the data that is output when you issue the LINKCONTROL command to obtain statistics relating to NS 3000/iX LAPBMUX Links.

LINKSTATE Parameter Fields

The following is an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=LINKSTATE command:

```
Linkname:  PLSPLNK7   Linktype:   PCI LAPBMUX  Linkstate:   CONNECTED LEVEL 2
```

Linkname — The Likname field specifies the name of the link.

Linktype — The Linktype field specifies the type of link, such as LAP-B, LAPBMUX, 100BT, or IEEE 802.3, that is being monitored, and the type of I/O bus it is on (will be preceded with "PCI" for PCI LAPBMUX links).

Linkstate — The Linkstate field specifies the current state of the link. The possible link states are as follows:

- Not connected
- Connected level 1.
- Connected level 2
- Connecting level 1.
- Connecting level 2.
- Disconnecting level 1.
- Disconnecting level 2.

In this example, in LINKSTATE Command for LAP-B Link, the current state of the LAPB link named **PLAPLNK7** is CONNECTED LEVEL 2.

The *LINKSTATE* parameter fields are displayed whenever you enter the LINKCONTROL Status command, regardless of which other parameters are specified.

CONFIGURATION Parameter Fields

The *CONFIGURATION* parameter for LAPBMUX links displays the *LINKSTATE* parameter fields and many additional fields. These additional fields display information that is related to the link configuration and which, except for the *Cable Type* parameter, are input through the NMMGR configuration program

LAPBMUX CONFIGURATION Parameter Output provides an example of the data that is displayed when you issue the LINKCONTROL

linkname;STATUS=CONFIGURATION command:

```
Linkname:  PLAPLNK7  Linktype:  PCI LAPBMUX  Linkstate: CONNECTED LEVEL 2

Physical Path:           0/6/2/1.7
Physical interface            V.35          LAPB parm T1 :sec              3
Transmission speed        20480009          LAPB parm N2 :retry           20
Clock source               External         LAPB parm K                    7
Local mode                      DTE         Module Count                   8
Connect timeout :sec            900         Buffer size :bytes          4096
Adapter timeout :sec             10
```

Physical Path — The Physical Path field displays the current physical path for the adapter card as specified in the NMCONFIG configuration file.

NOTE On PCI LAPBMUX links, the physical path generally has more components and is a “longer” path (e.g., 0/6/0/1.2, in which 0/6/0/1 is the path .2 is the port number) along with the port number associated with that link. The output for PCI links is different from that HP-PB.

Physical Interface — The Cable Type field displays the cable type that is currently connected to the PSI card. This data is provided by the PSI card and cannot be configured through NMMGR. The possible values for this field are as follows:

- RS232
- V.35
- None

Transmission Speed — The Transmission Speed field displays the current transfer rate, or clocking, configured for the node in the NMMGR Link screen. If modems are used, the modems will control the clocking. The ACC card transmits at the clocking setting of the modem and ignores the value configured in this field.

Clocking Source — The Clocking source field displays the clocking, configured for the node in the NMMGR Link screen. The possible types of clocking source are:

- External
- Internal

If modems are used, the modems will control the clocking and clocking source should be configured as “External”. If the node is connected to other node which requires clocking, then it should be configured as “Internal”.

Local Mode — The Local Mode field displays the value specified for local mode in the NMMGR link screen. The possible values for this field

are as follows:

- DTE, for connection to a device that is configured as a DCE.
- DCE, for connection to a device that is configured as a DTE.

Connect Timeout — The Connect Timeout field displays the current logical link level 2 connection timeout. The *Connect Timeout* parameter sets the amount of time a node will wait for a logical connection to a remote node to be established. If this timer expires, the node aborts the connection attempt. The abort process can take several additional seconds.

Adapter Timeout — The Adapter Timeout field displays the value specified for the Adapter timer. This timer, also called a heartbeat timer, is used to monitor whether the system and/or the ACC card are functioning. The ACC card, and the system, transmit a signal, called a heartbeat signal, to each other on a specified schedule. If, for example, the heartbeat does not arrive at the PSI card from the system (or vice-versa), the card or system waits the number of seconds specified by this field. If no heartbeat arrives before this timer expires, the link is dropped. You can determine whether the PSI card or the system failed by checking to see which device is still active. The default for this field is 60 seconds and it is recommended that you do not change the value of this field. The PSI always waits 20 seconds longer than the system waits before it drops the link.

LAPB Parm T1 — The LAPB Parm T1 field displays the current value of the T1 timer. The T1 timer waits the specified number of hundredths-of-seconds for a particular frame to be acknowledged. A frame that is transmitted, but not acknowledged, before the T1 timer expires, is retransmitted.

LAPB Parm N2 — The LAPB Parm N2 field displays the maximum number of times a frame is retransmitted after the LAPB Parm T1 expires. The frame is retransmitted at the LAPB Parm T1 interval for the number of times specified in this field. When this count is depleted, the frame is retransmitted at 20 second intervals. If no response is received after these transmissions, the link is brought down.

A node that is configured with the value specified in LAP-B *CONFIGURATION* Parameters output will attempt to retransmit an unacknowledged frame a maximum of 20 times at T1 intervals.

LAPB Parm K — The LAPB Parm K field displays the configured number of unacknowledged frames that are allowed in the network at any given time. For example, if this value is set to seven (7) for a node, and that node transmits seven (7) packet frames onto the network, it cannot transmit another frame until one or more of the transmitted frames are acknowledged.

Modulo Count — The Modulo Count field displays the maximum frame sequence number allowable for any frame in the network. This

field can be set to a value of 8 or 128, meaning that frames are numbered from either 0 through 7, or 0 through 127.

Buffer Size — The Buffer Size field displays the current buffer size. This value will be equal to the buffer size configured in the NMMGR Link screen plus 4 bytes of overhead that is added by the level 3 protocol.

STATISTICS Parameter Fields

The *STATISTICS* parameter for LAPBMUX links displays many fields in addition to the *LINKSTATE* parameter fields. The *CONFIGURATION* parameter fields are not displayed when this parameter is used. For an example of the data that is displayed when you issue the LINKCONTROL *linkname*;STATUS=STATISTICS command:

```
Transmit bytes          11543567876     Receive bytes           32948732576
TransmitS                  13166130     ReceiveS                   18338778
Transmits ctrl bytes              0     Receive ctrl bytes                0
Transmit ctrl                     0     Receive ctrl                      0
Transmit overruns                 0     Receive overruns                  0
Tx deferred: iova                 0     Recv deferred: iova               0
Tx deferred: buffer               0     Recv dropped: buffer              0
Level-2 connects                  1     Recv dropped: addr                0
Level-2 disconnects               0     Recv dropped: other               0
Connect timeouts                  0     Secs since clear             173980
```

This command displays statistics about data transmitted and received across the link. Many field values are summations. Over time, the values in these fields reach their maximum possible value. When this occurs, these fields can only be reset manually.

Transmit bytes — Total number of bytes successfully transmitted onto the medium. This includes normal data bytes as well as control data bytes. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 10.1 million bytes were transmitted, or about 9.7 Mbytes.

Transmits — Total number of frames successfully transmitted onto the medium at normal priority. Normal data as well as the control data frames. The byte count given by Transmit bytes is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example above, 127700 frames were transmitted at normal priority.

Transmit ctrl bytes — Total number of control bytes successfully transmitted onto the medium. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 0 control data bytes were transmitted.

Transmit ctrl — Total number of control frames successfully transmitted onto the medium at normal priority. The byte count given

by Transmit ctrl bytes is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 0 control frames were transmitted at normal priority.

Transmits overruns — Total number of frames that were deferred to other network traffic before their initial transmission onto the network. This means that the 100Base-T card had to wait for carrier to drop and stay dropped for a few nanoseconds, before attempting to transmit the frame. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 1738 transmit frames were deferred.

Level-2 connects — Number of times the link driver noticed the link had previously been established, but was no longer up. This may occur because the cable was unplugged, the hub was powered off, the hub automatically requested a reconnect, or normally (at link shutdown time). This total does not include repetitive, failed attempts by the link driver to reestablish the link. The maximum value of this 32-bit field is 2147483647.

Level-2 disconnects — Number of times the link driver noticed the link had previously been established, but was no longer up. This may occur because the cable was unplugged, the hub was powered off, the hub automatically requested a reconnect, or normally (at link shutdown time). This total does not include repetitive, failed attempts by the link driver to reestablish the link. The maximum value of this 32-bit field is 2147483647.

Receive bytes — Total number of bytes successfully transmitted onto the medium. This includes normal data bytes as well as control data bytes. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 10.1 million bytes were transmitted, or about 9.7 Mbytes.

Receives — Total number of frames successfully received from the card at normal priority. This includes normal data as well as control data frames. The byte count given by Receive bytes is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 127700 frames were transmitted at normal priority.

Receive ctrl bytes — Total number of control bytes successfully received from the card. The maximum printable value of this 64-bit field is 17 digits, or about 99 million Gbytes. In the previous example, 10.1 million bytes were received, or about 9.7 Mbytes.

Receive ctrl — Total number of control frames successfully received from the card at normal priority. The byte count given by Receive Ctrl bytes is distributed over this number of frames. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 0 control frames were received at

normal priority.

Receive overruns — Total number of frames that were deferred to other network traffic before their initial transmission onto the network. This means that the 100Base-T card had to wait for carrier to drop and stay dropped for a few nanoseconds, before attempting to transmit the frame. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 1738 transmit frames were deferred.

Recv dropped: buffer — Total number of frames received from the adapter card, but dropped because no data buffers were available from the upper layer protocol requesting to receive these frames. This is a relatively common occurrence, amounting to flow control for all protocols sharing those buffers. Many protocols include built-in mechanisms for detecting lost frames and requesting their retransmission from the remote side. The maximum value of this 32-bit field is 2147483647. In the example above, out of 66.4 million frames received, 262 have been dropped for lack of buffer resources; any similar level would be considered normal.

Recv dropped: addr — Total number of frames received by the adapter card, but dropped because no upper layer protocol had requested future reception of those frames, or because that protocol unbound itself from the link while the received frame was still in motion. Older, intelligent adapter cards can invisibly receive and drop these frames, often without ever reporting them as statistics. But today's adapters are not intelligent, and require link driver involvement. For users unfamiliar with seeing it, this statistic may seem excessive. But it is important because it gives an indication of the amount of unnecessary traffic present on the network segment to which the adapter card is connected. High values may indicate a need to resegment the network, since systems and their adapters are spending a large amount of time and resources recognizing and dropping frames they do not care to see. High volumes of such traffic can also limit network bandwidth. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, out of 15.7 million frames received, 9.87 million have been dropped based on address: about 63% of all frames received are then being dropped. This level would be considered high.

Recv dropped: other — Sum total number of frames received from the adapter card, but dropped because: an upper layer protocol error was returned; the required address format was not supported; the frame arrived while the link driver was in an unusual state; or for perfect multicast filtering reasons. The maximum value of this 32-bit sum is 2147483647. In the example above, 24785 frames have been dropped for one or more of the reasons listed; this number would be considered high, and further investigation might be needed if it appears to be impacting any applications.

Recv deferred — Number of times an address-matched receive frame was temporarily held (queued) by the link driver, because of a momentary lack of DMA resources. Once those resources became available, the frame was automatically transferred to the host. The maximum printable value of this 64-bit field is 17 digits, or about 99 million billion frames. In the previous example, 123 frames were temporarily held.

Secs since clear — The number of seconds elapsed since statistics were last reset via the LINKCONTROL *linkname;* STATUS=RESET command. This gives the sample time over which the displayed statistics have been collected. Per-time-unit figures may then be calculated if desired. The maximum value of this 32-bit field is 2147483647, or about 68 years.

RESET Parameter Fields

The *RESET* parameter for LAPBMUX links reset all the accumulated statistics for the links, The command also displays all of the *LINKSTATE*, *CONFIGURATION*, and *STATISTICS* parameter fields. Refer to the *STATISTICS* parameter for a description of the displayed fields.

ALL Parameter Fields

The *ALL* parameter for LAPBMUX links displays all of the *LINKSTATE*, *CONFIGURATION*, and *STATISTICS* parameter fields. This is an example of the *ALL* parameter output:

```
Linkname:  PLAPLNK7  Linktype:  PCI LAPBMUX  Linkstate: CONNECTED LEVEL 2

Physical Path:           0/6/2/1.7
Physical interface            V.35          LAPB parm T1 :sec              3
Transmission speed           56000          LAPB parm N2 :retry           20
Clock source              External          LAPB parm K                    7
Local mode                     DTE          Module Count                   8
Connect timeout :sec           900          Buffer size :bytes          4096
Adapter timeout :sec            10

Transmit bytes          11543567876     Receive bytes           32948732576
TransmitS                  13166130     ReceiveS                   18338778
Transmits ctrl bytes              0     Receive ctrl bytes                0
Transmit ctrl                     0     Receive ctrl                      0
Transmit overruns                 0     Receive overruns                  0
Tx deferred: iova                 0     Recv deferred: iova               0
Tx deferred: buffer               0     Recv dropped: buffer              0
Level-2 connects                  1     Recv dropped: addr                0
Level-2 disconnects               0     Recv dropped: other               0
Connect timeouts                  0     Secs since clear             173980
```

B Submitting an CR

For further assistance from HP, document the problem as an CR (change request) and forward it to your HP Service Representative. Include the following information where applicable:

- A characterization of the problem. Describe the events leading up to and including the problem. Attempt to describe the source of the problem. Describe the symptoms of the problem and what led up to the problem.

  Your characterization should include: MPE/iX commands; communication subsystem commands; job streams; result codes and messages; and data that can reproduce the problem.

  Illustrate as clearly as possible the context of any message(s). Prepare copies of information displayed at the system console and user terminal.

- Obtain the version, update and fix information for all software using `NMMAINT.PUB.SYS`. This allows Hewlett-Packard to determine if the problem is already known, and if the correct software is installed at your site.

- Record all error messages and numbers that appear at the user terminal and the system console.

- Run `NMDUMP.PUB.SYS` to format the NM log file that was active when the problem occurred (`NMLG`*`nnnn`*`.PUB.SYS`). You may need to issue the MPE/iX command `SWITCHNMLOG` to free the NM log file.

  Using NMDUMP, format the log file for NETXPORT (3), NETIPC (5), Network Services (6) and link manager (8) information. Inspect the formatted output and try to locate errors. Prepare the formatted output and a copy of the log file for your Hewlett-Packard representative to further analyze.

- Prepare a listing of the configuration file and the MPE/iX I/O configuration you are using for your Hewlett-Packard representative to further analyze. Inspect the output and try to locate errors.

- Try to determine the general area within the software where you think the problem exists. Refer to the appropriate reference manual and follow the guidelines presented in that manual.

- Issue the `LINKCONTROL` *`linkname`*`; STATUS=` command for each link. Retain the output for your Hewlett-Packard representative to further analyze.

- Document your interim, or “workaround” solution. The cause of the problem can sometimes be found by comparing the circumstances in which it occurs with the circumstances in which it does not occur.

- Create copies of any NS 3000/iX or NetIPC user trace, network transport trace and communication link trace files that were active

when the problem occurred for your Hewlett-Packard representative to further analyze.

- In the event of a system failure, a full memory dump must be taken.
- Make any NI DMP*xxx* files available for your HP service representative.

Index

X