

Burroughs 

Burroughs
Network Architecture
(BNA)

NETWORK CONTROL

REFERENCE MANUAL
VOLUME 2

PRICED ITEM

Burroughs
Network Architecture
(BNA)

NETWORK CONTROL

REFERENCE MANUAL
VOLUME 2

Copyright © 1981 Burroughs Corporation, Detroit, Michigan 48232

PRICED ITEM

“The names used in this publication are not of individuals living or otherwise. Any similarity or likeness of the names used in this publication with the names of any individuals, living or otherwise, is purely coincidental and not intentional.”

Burroughs believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this document should be addressed directly to Burroughs Corporation, Box CB7, Malvern, PA. 19355, Attn: Systems Documentation Dept., TIO East.

LIST OF EFFECTIVE PAGES

Page	Issue
Title	Original
ii, iii	Original
iv	Blank
v thru vii	Original
viii	Blank
1-1 thru 1-16	Original
2-1 thru 2-4	Original
3-1 thru 3-11	Original
3-12	Blank
4-1 thru 4-26	Original
5-1 thru 5-30	Original
6-1 thru 6-14	Original
7-1 thru 7-3	Original
7-4	Blank
8-1 thru 8-5	Original
8-6	Blank
9-1 thru 9-21	Original
9-22	Blank
10-1 thru 10-10	Original
A-1 thru A-9	Original
A-10	Blank
B-1 thru B-3	Original
B-4	Blank
C-1 thru C-6	Original
D-1, D-2	Original
E-1, E-2	Original
F-1 thru F-3	Original
F-4	Blank
G-1	Original
G-2	Blank
H-1	Original
H-2	Blank

TABLE OF CONTENTS

Section	Title	Page	Section	Title	Page
1	GENERAL	1-1	4	Tuning Attribute Commands	4-12
	General Command Description		(Cont.)	Routing Oriented Commands	4-12
	Format	1-2		Link Resistance to Neighbor Command	4-16
	Agents	1-4		Subport Communication Commands	4-17
	Operator Display Agents (ODT) Agents	1-4		Inter-Host Communication Commands	4-21
	Operations Interface Message (OIM) File Agents	1-4		Station-Oriented Commands	4-24
	Program Agents	1-4		Tunings Inquiry Command	4-25
	Types of Messages	1-5		Attribute Summary	4-26
	Classification of Commands	1-6	5	CONFIGURATIONAL COMMANDS	5-1
	Isolated Mode Commands	1-9		Configurational Command Symmetry	5-2
	Form of Commands from Agents	1-10		Network Definition Commands	5-3
	ODT Commands	1-10		Add Host Command	5-4
	NW Prefix	1-10		Add Node Command	5-4
	Dedicate Command	1-11		Delete Host Command	5-5
	Commands from OIM Input Files	1-11		Delete Node Command	5-6
	Initialization File	1-11		Station Level Commands	5-8
	Program Agents	1-12		Station Attributes	5-8
	Form of Responses and Reports to Agents	1-12		Station-Level-Manager Attributes	5-8
	ODT Agents	1-12		Optional Station Attributes	5-9
	OIM Printer (or Printer Backup) Messages	1-12		Required Station Attributes	5-10
	Program Agents	1-12		Optional Station Event-Monitor Attribute	5-12
	Handling of Operations Interface Messages	1-13		Optional Station Counter-Monitor Attributes	5-12
	ODT Agent	1-13		ACU-Station Attributes	5-13
	OIM Input Files	1-13		Station Commands	5-15
	Program Agents	1-13		Order of Entering Station Commands	5-15
	Negative Responses and Error Messages	1-13		Station Profile Commands	5-15
	Syntax Errors	1-14		Add Station Command	5-18
	Other Errors	1-14		Delete Station Command	5-20
	Response Steering	1-15		Modify Station Command	5-21
2	MODE COMMANDS	2-1		Ensemble Commands	5-22
	NET Command	2-1		Add Ensemble Command	5-22
	ENDINIT Command	2-4		Delete and Modify Ensemble Commands	5-24
3	ACCESS CONTROL COMMANDS	3-1		Neighbor Connection Commands	5-26
	Node Password Command	3-1		Add Connection Command	5-26
	User Authorization Command	3-4		Delete and Modify Connection Commands	5-29
	Validate Command	3-6		OPERATIONAL COMMANDS	6-1
	Program Agent Control Commands	3-8		Calls with Neighbor Commands	6-1
	Program Agent Associate Command	3-9		Await Call Command	6-1
4	ATTRIBUTE ENTRY COMMANDS	4-1	6	Clear Call Command	6-3
	Local Identity Command	4-1		Establish Call Command	6-4
	Get-Only Attributes	4-3		Save and Ready Station Commands	6-8
	Network Oriented Commands	4-5		Host-to-Host Communications	6-10
	Logging and Monitoring Commands	4-7			
	Logging Option Command	4-7			
	Monitor Traffic Command	4-9			
	Port Log Interval Command	4-11			

TABLE OF CONTENTS (Cont.)

Section	Title	Page	Section	Title	Page
6	Other Operational Commands	6-12	10	Station Reports	10-7
(Cont.)	Log Comment Command	6-12	(Cont.)	Validation Failure Reports	10-7
	Routing Refresh Command	6-12		Station Attach and Detach Report	10-7
	Start Trace Command	6-13		Station Failure Report	10-8
	Sending Text to an Agent Command	6-14		Test Reports	10-8
7	STATION CONFIDENCE (MANUAL) COMMANDS	7-1		Station Error Report	10-8
8	SERVICE COMMANDS FOR OIM INPUT FILES	8-1		Port Level Reports	10-9
	Load Command	8-1		Host Saved Report	10-9
	Stop Command	8-3		Port Level Error Reports	10-9
	Continue Command	8-4		Validation Failure Report	10-9
	Discard Command	8-5		Subport Dialog Establishment Failure Report	10-9
9	SUMMARY AND DETAIL INQUIRIES	9-1	A	Port Level Error Report	10-10
	Host Summary and Detail	9-1		OIM MESSAGE SYNTAX	A-1
	Host Summaries	9-2		Railroad Diagrams	A-1
	Host Detail	9-3		Railroad Component	A-2
	Node Summaries	9-4		Definition of Common Syntactic Elements	A-4
	Neighbor and Connection Information	9-4		Key Word Representation	A-7
	Neighbor Summaries	9-5		Time Intervals	A-8
	Neighbor Connection Information	9-6		BDLC Call Data <bdlc-call-data >	A-9
	Neighbor Detail	9-7	B	HOW TO PICK MAX HOP COUNT AND MAX RESISTANCE FACTOR (MHC AND MRF)	B-1
	Routing Information	9-8	C	GENERAL INDEX, ALPHABETICAL ORDER	C-1
	Responses for Destination Inquiries (TO < node >)	9-9	D	COMMAND INDEX, ALPHABETICAL ORDER	D-1
	Responses for Neighbor Routings (VIA < node >)	9-10	E	COMMAND INDEX MNEMONICS, ALPHABETICAL ORDER	E-1
	Profile Summary and Detail	9-12	F	ERROR MESSAGE INDEX, ALPHABETICAL ORDER	F-1
	Station Summary and Detail	9-13	G	REPORT INDEX, ALPHABETICAL ORDER	G-1
	Station Summaries	9-13	H	MINIMUM REQUIREMENTS FOR NETWORK OPERATION	H-1
	Station Detail	9-15			
	Ensemble Summary and Detail	9-17			
	Candidate Summary and Detail	9-18			
	Port Counts Inquiry	9-20			
	Mode Inquiry	9-21			
10	REPORTS	10-1			
	Network Services Manager Reports	10-1			
	Network Services Manager Phase Change Report	10-1			
	Message from a Remote Agent Report	10-2			
	Routing Reports	10-2			
	Router Frame Error Reports	10-2			
	Routing Trace Report	10-4			
	Destination Node Status Change Report	10-5			
	Station Level Reports	10-6			
	Neighbor Status Reports	10-6			

INTRODUCTION

This manual is intended for use with Burroughs Network Architecture (BNA) by programming and computer operations personnel. Additionally, the information can be used by network/system planning personnel who require detailed information of BNA operations.

It is assumed that the reader has a basic understanding of the operation of Burroughs computer systems and data communications, and has studied the Burroughs Network Architecture (BNA) Reference Manual, Volume 1, form 1132172.

This manual describes the interface messages that are associated with BNA Network Services. These messages are used to control, configure, and interrogate various BNA Network Services functions.

This is a general Burroughs systems document, not particular to any specific system. Where specific system information is required, the particular system publication should be referred to. This manual and any system publication should be considered as a set for a given system installation.

For the operator (ODT agent) this manual serves as a reference to all operational commands and responses; however, it must be used in conjunction with a specific system publication that describes the precise entry of commands and display formatting.

For the programmer who is writing Program Agent programs, this manual serves as a reference to all messages to or from the Operations Interface. However, this issue of the manual does not include the encoded detail for the messages. The encoded detail will be provided in a later revision. Meanwhile, the manual provides valuable Program Agent planning information.

Generally, this manual is divided into logical functions, closely following the modes of a BNA node. A brief synopsis of the sections follows:

Section 1 deals with the general topics of the Operations Interface and the commands and responses that traverse this interface. It also includes an introductory discussion on the general layout and format of the manual.

Sections 2 through 9 describe the detailed syntax and semantics of commands and responses. However, unlike many operation manuals, this manual groups the commands and responses into their functional classifications rather than a collated sequence. A comprehensive index is provided at the rear of the manual that is ordered alphabetically and can be used to locate commands, both in the complete form and in the mnemonic form. Other indexes are also provided to aid in locating other items of interest.

Section 10 describes the formats of the unsolicited responses (Reports).

The Appendixes contain reference information for the use of this manual, such as indexes, railroad, and syntax rules, and miscellaneous operations related subjects.

SECTION 1

GENERAL

The Operations Interface is the means whereby an agent external to BNA, such as a human operator, a program, or a file, can exchange control information with a BNA node. These messages can be used by the external agent:

1. To initialize the node into the network.
2. To make inquiries about the status of the tables and attributes.
3. To define and modify the node's network configuration.
4. To change certain operating parameters.
5. To initiate a specific action.
6. To be informed of and to handle exception conditions.
7. To remove (shut down) the node from the network.
8. To perform testing and maintenance.

This manual describes the command and response messages that are used to monitor and control a BNA node within a BNA network. The messages, which are called Operations Interface Messages (OIMs), have message formats that are the same for all systems. However, there are also some system-dependent features and limitations that must be considered for actual operating procedures. For details of the features and limitations of a system, refer to the publication for that particular system.

For system operation, both this manual and the system-dependent manuals should be consulted. For planning purposes, this manual provides sufficient information to understand the operating features of BNA.

NOTE

In designing the network architecture, a great effort was made to make all BNA systems conform to an identical set of operating messages. In general this goal was met. However, there will be some differences, particularly with responses. Therefore, although this manual attempts to describe output messages (responses) in precise detail, some differences in responses can be expected. These differences involve the type of format or actual content, but the response will be such that an operator can readily recognize the response intent. For Program Agents, the encoded responses (to be provided in a later issue) will be precise; no interpretation of intent will be needed.

As previously mentioned, this manual is organized logically, but within logical groupings, items are ordered alphabetically, where possible. Where the reader cannot easily find a specific item, a comprehensive index is provided in Appendix C.

The following subsection describes the general order and layout of the message descriptions used in this manual.

GENERAL COMMAND DESCRIPTION FORMAT

The general order of command descriptions presented in this document is:

- . command syntax
- . semantics
- . servicing phase
- . default values
- . examples
- . response syntax
- . error messages

In those instances where little detail is needed, only relevant parts are included. A brief description of each follows:

Command Syntax

The command syntax describes the precise format by use of railroad syntax diagrams. Refer to Appendix A for railroad rules and common syntactic elements.

Semantics

A brief description of the command and its effect is given.

Servicing Phase

This part describes any restrictions as to when the command can be entered.

Section 1
General

Default Value

When attributes have predefined default values, the values are listed.

Examples

Examples of the form of the commands, and in some cases the form of the positive response, are given for each command.

Response Syntax

The syntax for the positive response is described by use of railroad diagrams.

Where the inquiry form of command results in a different response, both response constructs are shown under the subtitles of "command response syntax" and "inquiry response syntax".

Error Messages

For every command there is a corresponding set of possible error conditions. With the exception of those described under Negative Responses and Error Messages in this section, all error messages are shown with the command. To aid in finding error message meanings, error messages are indexed in Appendixes at the end of this manual.

AGENTS

Agents can be operators, programs, or files that exchange messages with a BNA node. The types of agents and their operations are described in the following paragraphs.

OPERATOR DISPLAY TERMINAL (ODT) AGENTS

ODT agents are external agents that can be human operators working through a console. ODT agents can be local or remote to a node:

1. Local agents are the local host operating system console and any "remote" station defined as an ODT:
2. Remote agents are agents operating from input devices attached to a remote node.

OPERATIONS INTERFACE MESSAGE (OIM) FILE AGENTS

OIM file agents are disk files that contain Operations Interface Messages. Their operation is controlled by Service commands. Refer to Section 8 for further details on Service commands.

PROGRAM AGENTS

Program Agents are user written programs which can send and receive Operations Interface Messages.

Program Agents communicate with the Operations Interface through a Network Services port. They become Program Agents by nature of their function and their suitable security qualifications. Refer to Section 3; Program Agent Control.

The agent can be local to the Node being controlled, or remote, in which case the BNA network itself is used to establish and maintain this remote communication.

TYPES OF MESSAGES

The three types of Operations Interface Messages are Commands, Responses, and Reports. A description of each follows:

1. Commands are OIMs originated by the agent and sent to the BNA node. These messages are used to inquire about the state of the node, to change the operating behavior of the node (by changing the values of attributes), to define and modify the node's network configuration, and to initiate some activity within the node.
2. Responses are OIMs which are generated by the node as a direct result of receiving a command, and these are sent to the agent. Responses are positive if the intent of the corresponding command was successfully completed within the node. Responses are negative if the command was erroneous or the command could not be honored for any reason.
3. Reports are unsolicited messages sent by the node to the agent, usually indicating the occurrence of some event in the node.

Additionally, there are messages that are functional requirements for BNA Network Services but are not members of the OIM command set. Messages that fall under this category are HN (Host Name), and AT (At remote host).

HN (host name)

The HN function identifies the host name of the local system which supports BNA. The host name is supplied during system initialization.

While any user program is running, or any ports are open, the node host name identity cannot be changed.

```
HN ----- <host-name> -----|
    |                               |
    |-----|
```

Examples

```
HN
=>  HOSTNAME = "RED"
```

```
HN GREEN
=>  HOSTNAME CHANGED FROM "RED" TO "GREEN"
```

AT (AT host ODT input)

AT <hostname> <input message>

The AT command is recognized by the local host as an ODT input message that is destined for a remote host. The local host does not check the message, but passes the message to Host Services for onward transmission to the remote host. The remote host acts upon the message, as though it were received from a local ODT subsystem, and returns any responses to the originating host.

Examples

Refer to specific system documentation for examples of this command.

CLASSIFICATION OF COMMANDS

The Operations Interface Commands are grouped according to the class of function which they perform. Certain classes of commands can be entered at certain phases of the node operation.

The classes of commands, their general function, and the phase of the node during which they can be entered, are shown in figure 1-1 and are described in the following paragraphs.

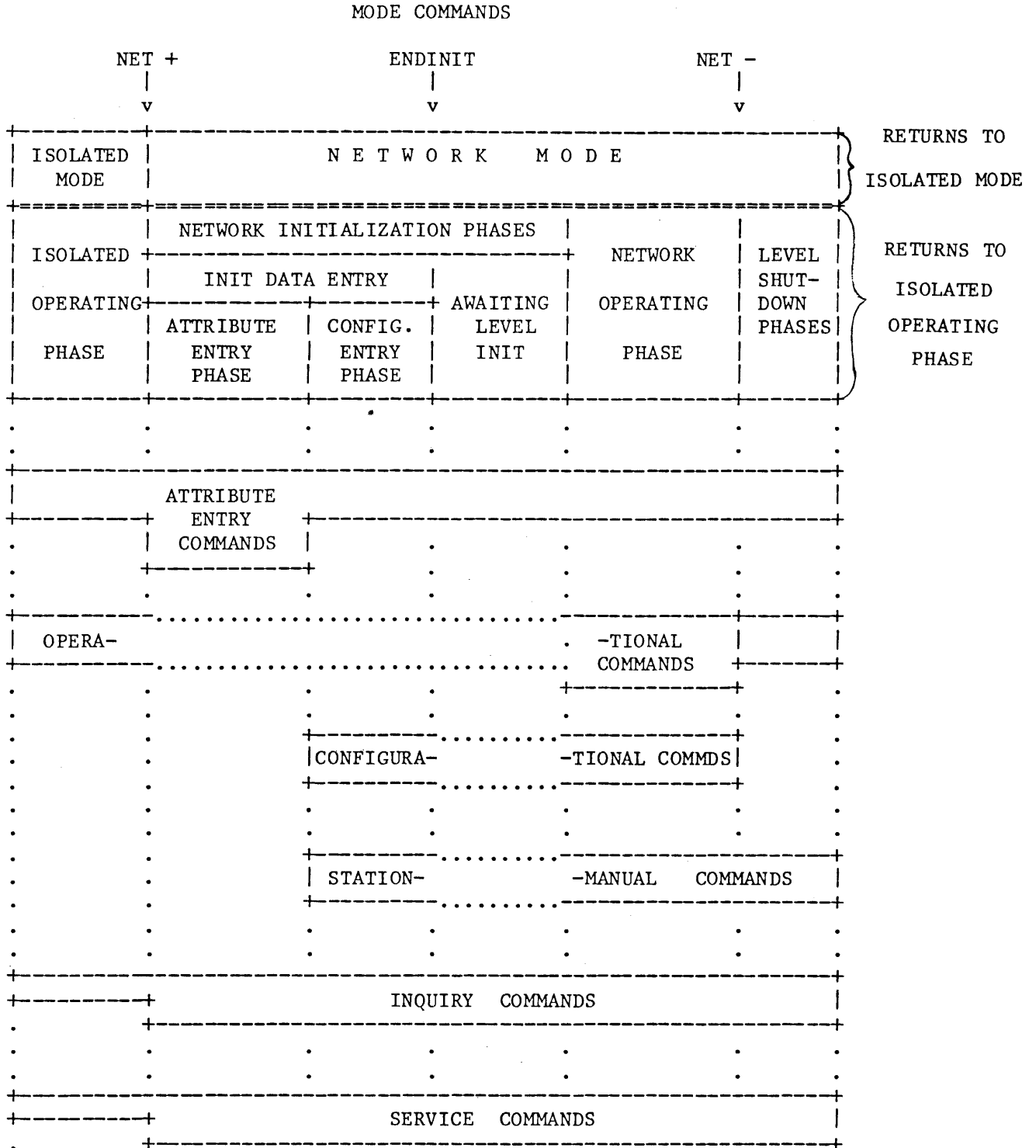


Figure 1-1. Classification of Operations Interface Commands

Mode Commands

Mode commands cause mode or phase changes of the node. The commands NET and ENDINIT are used to move from Isolated mode to Network mode and to start the Network shutdown phase of the node.

Attribute Entry Commands

Attribute entry commands set the Network Services attributes in the various Network Services Managers. These attributes are used during the initialization phase of the node to declare the node identity, to select certain operating options, to supply values for parameters which "tune" the performance of the node, and, in some implementations, to set the limits for resource utilization.

Most of these attributes possess default values, so that they need not be set; a few are required to be set during Network initialization.

After Network initialization, during the operating phase of the node, some of these attributes can be changed. A few can also be changed during the Isolated operating phase.

Configurational Commands

Configurational commands generate and change the network configuration information within the node. They define the hosts and nodes with which the local node is to communicate, and they set up information used to connect to neighbor nodes, defining a set of stations and their attributes.

These commands are operationally inert; they neither initiate nor terminate communications and connections. They simply set up information which then can be used by the Operational Commands to perform such functions.

Configurational commands can be entered in Network mode any time after the Attribute Entry phase of the node, except during shutdown. In fact, the entry of the first Configurational command indicates the end of the Attribute Entry and the beginning of the Configuration Entry phase of the Level Managers.

Operational Commands

Operational commands can be entered only during the operating phases of the node. In Network mode, these commands cause the local node to act relative to the network: to initiate or terminate communications with and connections to other nodes in the network, and to make or break associations with Program Agents.

Station Manual Commands

Station Manual commands are used to establish confidence in the proper operation of the Stations. Together with inquiry commands, they can be used to diagnose problems. They can be entered at any time after the Attribute Entry phase of Network initialization.

Inquiry Commands

Inquiry commands have no effect on the activity or performance of the node. They only retrieve information, and can be entered at any time. However, in Isolated mode, many of these commands have no meaning and are rejected if entered.

Inquiry commands are the most numerous commands and are used for a wide range of purposes, from very general, summary-type information gathering, to very detailed examination of attributes for fault isolation.

Service Commands

Service commands control the activity of OIM input files and handle the disposition of responses to commands. They can be entered at any time.

ISOLATED MODE COMMANDS

When the node is in Isolated mode, that is, prior to Network Services initialization, most Operations Interface commands are not serviceable. The response to entering such commands is: "NOT IN NETWORK MODE", and the command is rejected.

There are a few, however, which are meaningful. These are listed below; their syntax and semantics are described later in this document.

Attribute Entry:

PORT LOG INTERVAL

Mode Change:

NET ("NET = ", "NET + " forms can be entered)

Configurational:

No commands of this class can be entered.

Operational:

No commands of this class can be entered.

Inquiry:

HN (host name), PORT LOG INTERVAL, CANDIDATE, NET.

FORM OF COMMANDS FROM AGENTS

The syntax of OIMs is the same for ODT and File Agents; Program Agents use an encoded form of the same messages. Although similar, each kind of agent possesses certain characteristics which affect the form of the OIM. These are described in the following paragraphs.

ODT COMMANDS

When an agent uses a medium through which system input messages (system ODT messages) are also entered, the OIM messages form a subset of these ODT messages. To distinguish OIM Commands from other system input messages, either the OIM Commands are prefixed with "NW" and followed by at least one space, or the ODT is dedicated for network operations. A description of the NW prefix and the DEDICATE command follows:

NW Prefix

Command Syntax

```
NW --- <oim command> -----|
```

There are, however, several exceptions to this rule:

1. The HN (host name) command, the AT command, the PORT LOG INTERVAL command, the CANDIDATE command, and all forms of the NET command must not use the NW prefix.

2. ODT agents which are dedicated do not use this prefix while operating in this way. An ODT agent can become dedicated by using the DEDICATED command.

NOTE

The mode of the host, whether Isolated or Networking, has no bearing on this rule.

Dedicate Command

```
DEDICATED ----- + -----|
---          | - - -----|
```

Semantics

For the ODT agent entering this command, it turns on or off the requirement for prefixing an OIM command with NW. This command can be entered only by ODT agents.

COMMANDS FROM OIM INPUT FILES

The OIM input file is composed of a series of records of 72 or more characters in length. The first 72 characters form the text field. Characters 73-80 can be either not present, all blank, or strictly numeric; if numeric, they are treated as a sequence number. Any additional characters in the record are not used.

Commands are separated by semicolons. A single command can continue from one text field of a record onto the text field of the next sequential record, provided that no words or numbers are split from one record to another. In fact, the end of a record serves as a delimiter to separate words, just as spaces and special characters do. A percent sign (%) indicates that the remainder of the record is a comment. The percent sign is treated just as though the end of record has been reached.

INITIALIZATION FILE

The initialization file is a special case of the OIM input file. It is read during the Network Initialization phase of the node. It is identical in format to any other OIM file, and contains commands which are meaningful during the initialization of the node into Network mode.

PROGRAM AGENTS

The form of commands from Program Agents, communicated by means of the Operations Interface Handler Port and Subport, are encoded. This issue does not provide the syntax for the Program Agent communication; later revisions will provide the syntax when it is available.

FORM OF RESPONSES AND REPORTS TO AGENTS

Every command has associated with it a response, either positive or negative. In addition, there are reports that are unsolicited responses from Network Services levels. These form the output messages of Network Services Operations Interface.

Throughout this manual the responses to commands are described with the command syntax, both for the inquiry form of the command and for the response to the action of a command. Additionally, Section 10 describes unsolicited messages, called reports.

Responses are normally returned to the agent entering the command. Reports are normally returned to an ODT agent. Options that allow diverting and inhibiting responses and reports are described in the Response Steering subsection (later in this section).

Reports and responses to agents take on different forms for different agents. The following paragraphs describe these forms.

ODT AGENTS

Responses and Reports appear as described by the syntax. However, some systems prefix output messages with origin information (such as mix number), or simply an indication which visibly differentiates between input and output images. In general, the visual images of the OIM output messages are consistent with other output messages in a particular system.

OIM PRINTER (OR PRINTER BACKUP) MESSAGES

Responses can be diverted to a printer or printer backup file. Responses sent to this file are formatted in the same way as those sent to any other agent; they vary only in accordance with the record size property of the medium used. For details, refer to Response Steering subsection.

PROGRAM AGENTS

Each response or report is contained in a single message communicated through the agent's Port and Subport. The form of the response is encoded. Details of the encoding will be provided in later revisions of this manual.

HANDLING OF OPERATIONS INTERFACE MESSAGES

For each command entered through the Operations Interface, an immediate response, positive or negative, is returned. How commands are sequenced, and where responses and reports are sent, depend on the agent.

ODT AGENT

Commands entering through the ODT interface are accepted one at a time, and a response is given before the next command is accepted. The ODT subsystem (which is outside of Network Services, and within the operating system) can provide for queueing input messages and for manipulating that queue in the event of errors.

Responses to commands entered through an ODT are returned to that ODT. However, it is possible for the ODT agent to specify that responses be sent to a printer.

Reports from the node are normally sent to the ODT agent. Responses caused by initialization file commands are sent to the ODT agent.

OIM INPUT FILES

Commands entering through an input file are received in the order they appear on the file, until: (1) the end of the file is encountered, (2) a command results in an error, or (3) a STOP command is entered by the ODT agent or encountered in the file itself. A syntax error causes the loading of the file to be aborted; any other negative responses suspend the file.

If an ENDINIT command is encountered in an initialization file, and there are additional commands appearing on that file after the ENDINIT, the file reading is temporarily suspended until the node phase change to Network Operating mode is complete.

Except for those responses which are explicitly diverted or inhibited, responses to commands from OIM files are sent to the ODT agent.

PROGRAM AGENTS

Commands are received from the Program Agent, and responses to these commands are returned to the program agent. The next command is accepted only after a response to the previous command has been returned. Errors do not terminate communication with the Program Agent.

NEGATIVE RESPONSES AND ERROR MESSAGES

Negative responses are generated for commands which are syntactically invalid or which cannot be serviced for any reason. Negative responses are formatted as follows: *** <error message> : <copy of oim command>

SYNTAX ERRORS

A syntax error is any error which is detectable by scanning the command without having to service it or to know by what agent or during what mode the command is entered.

Examples of syntax errors are:

1. Incorrect keywords.
2. Violation of the sequence of syntactic elements as specified by the railroad diagram.
3. Syntactically-improper images for names, integers, strings, time intervals, file titles, and so forth.
4. Improper use of syntactic marks, such as commas, parentheses, brackets, colons.
5. Incorrect attribute names.
6. Incorrect values for attributes which have enumerated mnemonic representations.

The syntax error message describes, as closely as possible, the nature and location of the error.

A syntax error encountered in reading an OIM input file causes the reading of that file to be aborted (rather than simply suspended, as other errors do).

NOTE

Because syntax errors apply to every OIM command, they are discussed here in detail, but are not repeated with the description of each command.

OTHER ERRORS

Unlike syntax errors, all other errors are detectable only by knowing in what phase of the node the command was entered, the identity of the agent, or by servicing the command. Certain errors apply to nearly all commands. Therefore, instead of repeating them with the description of each command, they are presented here:

UNAUTHORIZED AGENT

- the agent entering that command is not authorized to do so.

NOT IN NETWORK MODE

- a network command is entered while in isolated mode.

IMPROPER PHASE

- the command cannot be entered in the current phase of the node.

ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET

- if the Attribute Entry phase is exited, and all the required attributes have not been set. The required attributes are: LOCALIDENTITY, plus any system-specific ones. The Attribute Entry phase is exited by entering the first Configurational command, or by entering the ENDINITIALIZATION command.

The following general error messages apply to many Inquiry Commands:

INVALID NODE ADDRESS

NO SUCH <entity>

- the entity referenced does not exist. These are:

host	node	usercode
station	ensemble	profile
file	agent	using process

Note that some Inquiry commands have no meaning while the node is in Isolated mode. When this happens, negative responses are not returned (they do not suspend the reading of an OIM input file, for example); instead the no-op type of positive response "NOT IN NETWORK MODE" is returned.

RESPONSE STEERING

Responses are returned to the agent entering the command associated with that response. An exception to this rule is for OIM input files (which are input only); in this case responses are sent to the ODT agent.

However, an agent can specify that the response to a command be diverted to a printer, or be inhibited (not returned at all). This is true for all responses except syntax errors. Syntax errors can neither be diverted nor inhibited; they are unconditionally returned to the agent entering the command. If syntax errors are encountered by commands entered from an input file agent, the error messages are unconditionally sent to the ODT agent.

To invoke alternate response options, the following can be appended to any command:

```
--- <oim-command>-----|
      | - : --- PRINTER --- |
      |   | -              - |
      |   | - INHIBIT --- |
      |   | -              - |
```

There exists one printer file for all agents, and it can be released for printing by the following command:

```
RELEASE --- PRINTER -----|
---      -
```

Reports can be received by all authorized agents (refer to Access Control Commands). Normally, the system ODT is designated, by default, to receive all reports. Any other agents are required to request reports by entering the REPORTS command. Note that only authorized agents, authorized to receive reports, will be granted reports.

To turn on or turn off the receiving of reports by an agent, the following command is entered:

```
REPORTS -----|
---      |
      | ----- + ----- |
      | ----- - ----- |
```

The agent entering this command can apply this option only to itself, and not on behalf of any other agent. By default, an ODT agent receives reports; a Program Agent does not.

REPORTS without the + or - sign inquires about setting of this option. The response is either REPORTS + or REPORTS -.

SECTION 2 MODE COMMANDS

The Mode commands move the node to or from the Network mode.

NET COMMAND

Command Syntax

```

NET -----|
---
|
| - = ----- <init-file> -----|
|
| - + -----|
|
| | - <init-file> -----|
|
| -----|
|
| | - NOW -----|
| | ---|
| | - CANCEL -----|
| | -----|

```

```

where <init-file> is: ----- <file-title> -----|
|
| - *NULL -----|
| | ---|
| - *DEFAULT -----|
| | -----|

```

NOTE
 For the inquiry form of the command, refer to Section 9,
 Summary: NET.

Semantics

NET = <init-file> form:

The file information replaces the current value of the NEXT INIT FILE title. The next time the node is initializing into Network mode, it will use the information supplied by this command.

If <file-title> is supplied, this file becomes the NEXT INIT FILE; the next NET + command will use this for initialization. If *NULL is specified, no initialization file is automatically used; some other agent must supply the required commands for the node initialization into Network mode. If *DEFAULT is supplied, the <init-file> value reverts to the system initial value.

NET + form:

This specifies that the node is to move from Isolated mode into Network mode. The Network initialization proceeds, and the NEXT INIT FILE is used.

NET + <init-file> form:

This produces the same effect as entering NET = <init-file>, followed by NET +.

NET - form:

This specifies that the network operation is to be shut down, and thus the mode is to change from Network to Isolated. This shut-down is to take effect after the deactivation of all local subports that were in communication with remote ports.

NET - NOW form:

This is the same as "NET -", except that all subports in communication with remote hosts (ports) are deactivated immediately.

NET - CANCEL form:

This rescinds a previous NET - command, provided that the Port Level has not yet completed its Network mode shutdown.

Servicing Mode

The NET = form (with just file information) or NET (inquiry) can be entered at any time. The NET + form can be entered only in Isolated mode; the NET - form can be entered only in Network mode. The NET - CANCEL command can be entered only while a NET - or NET - NOW command is being serviced.

Default Value

The NET + form must be entered in order for the node to go from Isolated mode into Network mode. However, where a system can specify that the node be initialized directly into Network mode, NET + need not be used. All other forms of the NET command are optional.

Response Syntax

for NET + => INITIALIZING NETWORK MODE

for NET - => SHUTTING DOWN NETWORK MODE

Error Messages

SHUTDOWN CANNOT BE CANCELLED

- if the NET - CANCEL command arrived too late to be honored.

ENDINIT COMMAND

Command Syntax

```
ENDINITIALIZATION -----|  
-----
```

This command declares the end of the data entry phase of the node's initialization into Network mode. If ENDINIT appears in a file, and other commands follow it in that file, reading of the file is suspended until the phase change caused by ENDINIT is completed.

Servicing Mode

ENDINIT can be entered only in the Initialization Data Entry phase of Network mode.

Default Value

The ENDINIT command is required.

Response Syntax

```
INITIALIZATION DATA ENTRY PHASE COMPLETE
```

Error Messages

```
ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET  
- if ENDINIT is entered before the LOC and HN attributes required for  
Network initialization have been entered.
```

SECTION 3 ACCESS CONTROL COMMANDS

The following access control commands are used to control access to node resources, and agent privileges.

NODE PASSWORD COMMAND

NOTE

This command applies only to systems which permit the entry of Node Passwords through the Operations Interface.

Command Syntax

```

NODEPASSWORD --- <host-name> ----->
- - - - - |                               | -- #<node-address> -- |
           |                               |                               |
           |                               | -- #<node-address> ----- |
  
```

```

|<----->|
>----- MYNODEPASSWORD ----- = --- <string> -----|
| - YOURNODEPASSWORD - - - - - | | - HEX <hex-number> - |
| - - - - - | | - *NULL ----- |
| - - - - - | | - - - - - |
  
```

password range: <string>, 1 to 17 characters
or <hex-number>, 1 to 17 hex digits pairs.

Semantics

This command is used to add or delete the passwords that nodes send to each other during greeting processes.

NODEPASSWORD <node> MYNODEPASSWORD = <password>:

This form of the command sets or changes the password which is sent by the local node to the specified node in the Station Level and Port Level greetings.

NODEPASSWORD <node> YOURNODEPASSWORD = <password>:

This form sets or changes the password that the local node expects from the specified node in greeting messages.

Passwords are deleted by entering "*NULL" in place of the password.

where <password> is defined as:

```
>--- <string> -----|
|                       |
| - HEX <hex-number> - |
|   ---                |
| - *NULL -----     |
|   ----                |
```

NOTE
----- THERE IS NO INQUIRY FORM FOR THIS COMMAND -----

Default Value

The default value for all passwords is null (that is, no password). Null passwords can be explicitly entered by using the delete form of the command or by entering *NULL in place of the password.

Examples

```
NODEPASSWORD RED #2 MYNODEPASSWORD = ZZ YOURNODEPASSWORD = YY
NPW RED #2 MNPW = ZZ YNPW = YY
NPW BLUE MNPW = "##abc" YNPW = HEX 99AABB
```

Response Syntax

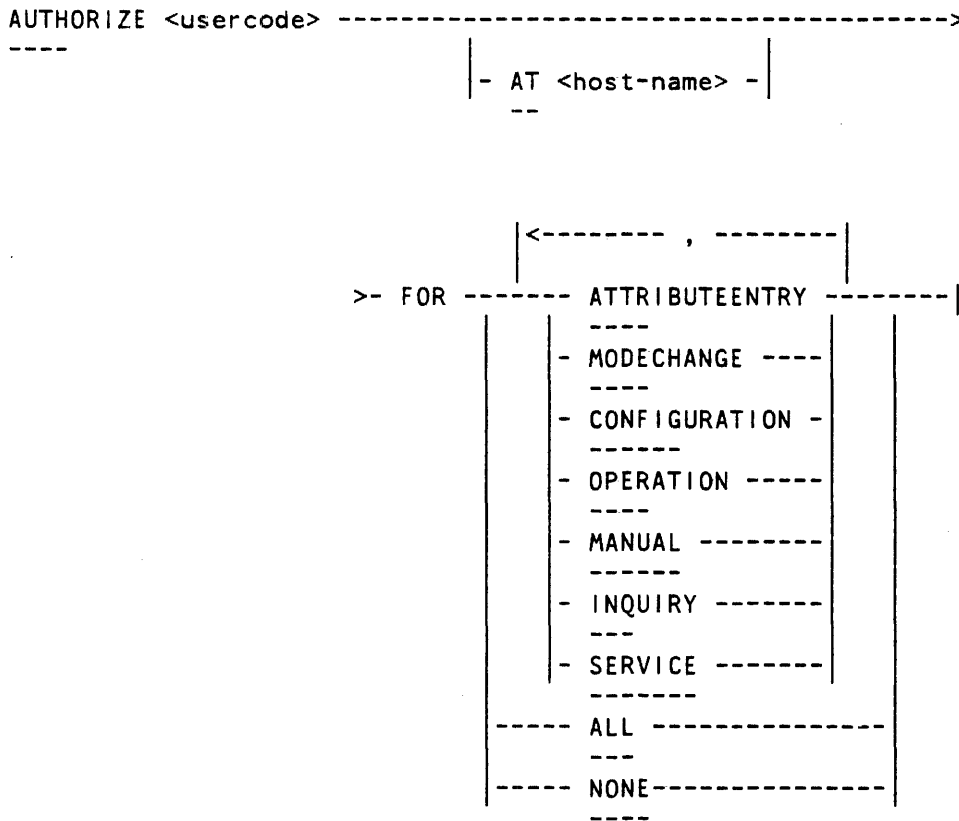
```
NODEPASSWORD CHANGED FOR <host-name> #<node-address>
```

Error Messages

```
ILLEGAL PASSWORD
  - passwords exceeds range of characters or digits.
NO SUCH HOST OR NODE
```


USER AUTHORIZATION COMMAND

Command Syntax



Semantics

This command authorizes a user (usercode) to enter certain classes of Operations Interface commands, or removes that authority. The agent which grants or removes the specified authorization must possess that authority itself.

If <host-name> is not specified, the local host is assumed.

This command overrides the effect of any previous authorize command. It takes immediate effect, whether or not the user is attached as an agent.

Servicing Phase

This command can be entered at any time during Network mode.

Default Value

This command is optional; the default value for any usercode is INQUIRY. There are two exceptions to this: (1) at least one agent (for example, the BNA system supervisor) has the authority of ALL, and (2) the initialization file agent (if this agent possesses a different usercode from the system operator) also has the authority of ALL.

Examples

AUTHORIZE CONTROLUSER FOR ATTR, CONFIG, OPER, INQ

Command Response Syntax

<usercode> AUTHORIZED FOR <option as specified in command>

Inquiry Response Syntax

```
=> AUTHORITY FOR <usercode> ----->
      | - AT <host-name> - |
      |
      | <----- , -----> |
      |----- ATTRIBUTEENTRY -----|
      |-----|
      | - MODECHANGE -----|
      |-----|
      | - CONFIGURATION -|
      |-----|
      | - OPERATION -----|
      |-----|
      | - MANUAL -----|
      |-----|
      | - INQUIRY -----|
      |-----|
      | - SERVICE -----|
      |-----|
      |----- ALL -----|
      |-----|
      |----- NONE -----|
      |-----|
```

*Error Messages***INVALID AUTHORIZATION OPTION**

- if option is other than those specified in command syntax.

INVALID USERCODE

- if usercode does not exist.

AUTHORITY CANNOT BE GRANTED OR REMOVED

- if user entering command specifies an authority it does not possess.

VALIDATE COMMAND*Command Syntax*

```

VALIDATE -----|
---
| - - - - - HOST -----|
| -----|
| - NEIGHBOR -----|
| -----|
| - HOST, NEIGHBOR -----|
| -----|
| - NONE -----|
| -----|
| - ALL -----|
| -----|

```

Semantics

This command supercedes any previous VALIDATE command.

This command sets the validate options in the various levels. If HOST is specified, the initiation of communications with any other host occurs only if the name of that host appears as a valid remote host.

If NEIGHBOR is specified, each neighbor must be validated before a station call with that neighbor can succeed. (If NEIGHBOR VALIDATION is false, validation can still occur on an individual neighbor when the appropriate connection information is supplied; refer to ADD CONNECTION command.)

If ALL is specified, the combination of HOST and NEIGHBOR validation is used, plus validation at the Router level. If Router validate is true, only those nodes which are known to the Router are accepted.

This command sets the VALIDATE attribute in the level managers as follows:

Command option	Port level	Router	Station level	
VALIDATE = NONE	F	F	F	
VALIDATE = NEIGHBOR	F	F	T	
VALIDATE = HOST	T	F	F	(T = TRUE,
VALIDATE = HOST, NEIGHBOR	T	F	T	F = FALSE)
VALIDATE = ALL	T	T	T	

NOTE

It is not possible to set the Router validate to true without also setting those of the Port and Station Levels.

Servicing Phase

These can be entered only during the Network Mode Attribute Entry phase.

Default Value

This command is optional. The default value is NONE.

Examples

VALIDATE = ALL
VAL = HOST, NEI

Response Syntax

```
VALIDATE = --- NONE-----|
          | - NEIGHBOR -----|
          | - HOST -----|
          | - HOST, NEIGHBOR ---|
          | - ALL -----|
```

Error Messages

VALIDATE OPTION INVALID
- if command options are other than those specified.

PROGRAM AGENT CONTROL COMMANDS

The following commands control the Program Agent access to the Operations Interface of a local node.

NOTE

Later revisions of this manual will provide additional information concerned with the operation and control of these agents.

Command Syntax

```
MAXPROGRAMAGENTS -----|
-----|
          | - = -- <integer> -----|
```

<integer> range: 0 to 65535

```
PROGRAMAGENTSECURITY -----|
-----|
          | - = --- PRIVATE -----|
          |   -----|
          | - PUBLIC -----|
          |   -----|
          | - GUARDED BY <file-title> -|
          |   -----|
```

These commands set the maximum number of Program Agents which can be connected during the operating phase of the node, and the type of security to be used.

To initiate communications with a Program Agent, refer to the PROGRAM command.

Servicing Phase

These commands can be entered only when the node is initializing to or operating in Network mode, and when there are no connected Program Agents (that is, there are no Subports open for the Operations Interface Handler Port).

Default Value

These commands are optional. The default value for MAX PROGRAM AGENTS is zero, which means that no Program Agents can be connected. The default for PROGRAM AGENT SECURITY is PRIVATE.

Examples

```
MAXPROG = 2  
PROGSEC = GUARDED BNA/GUARDFILE
```

Response Syntax

```
MAXPROGRAMAGENTS = <integer>  
  
PROGRAMAGENTSECURITY = --- PRIVATE -----|  
                        | - PUBLIC -----|  
                        | - GUARDEDBY <file-title> -|
```

Error Messages

```
INVALID MAX PROGRAM AGENT VALUE  
  - if integer outside allowable range.  
  
INVALID PROGRAM AGENT SECURITY OPTION  
  - if option is other than that specified in command syntax.
```

PROGRAM AGENT ASSOCIATE COMMAND

Command Syntax

```
PROGRAM -----|  
-----|  
      | - - - - <agent-name> -----|  
      | - + - - - <agent-name> = <prog-agent-specif.> --|
```

where <program-agent-specif.> is:

```
----- <your-name> ----->  
      | - *ANY -----|      | - AT --- <your-host> -|  
      | -----|      | --|  
      | -----|      | - *ANY -----|  
      | -----|      | -----|  
>-----|  
      | - , USERCODE = <usercode>-|
```

Semantics

These commands establish or break communications with a Program Agent. PROGRAM is an operational command, and as such, it can only be entered during the operating phase.

The PROGRAM + command:

This command initiates communication with a Program Agent. The <agent-name> is a name used to specify an Operations Interface subport used for this purpose.

The <your-name> is the YOUR NAME value of the Operations Interface Handler's specified subport. If *ANY is used, the Operations Interface subport connects to any Program Agent's MY NAME.

The <your-host> is the value of the subport YOUR HOST. If AT <your-host> is omitted, the local host is assumed. If *ANY is used, any host will be accepted.

The <usercode> is the YOUR USERCODE attribute value of the specified Handler's subport. If the Program Agent security is PUBLIC or GUARDED, the usercode must not be specified. If the security is private the usercode can be specified; if it is not, the usercode of the agent entering this command is assumed.

The PROGRAM - command:

This command breaks the communication with the Program Agent. The subport corresponding to the <agent-name> is closed abruptly.

Servicing Phase

Program Agents can be associated (connected) only during the operating phase of Network mode. Should the node return to Isolated mode, associations with all Program Agents are broken.

PROGRAM AGENT SECURITY and MAXPROGRAMAGENTS commands control the accessibility and visibility (that is, security requirements) of the Program Agents.

Examples

```
PROG + CONTROLLER = BNASUPERVISOR, UC = SITE
PROG + CONFIGINFO = *ANY AT MAINHOST
PROG - CONFIGINFO
```

Command Response Syntax

```
<agent-name> ASSOCIATED AS PROGRAM AGENT
```

```
<agent-name> DISASSOCIATED AS PROGRAM AGENT
```

Inquiry Response Syntax

```

      |<-----,-----|
=> PROGRAM ----- <agent-name> = <agent-info> -----|
```

where <agent-information> is:

```
----- <your-name> ----- AT ----- <your-host> ----->
| - *ANY ----- |           | - *ANY ----- |
>-- ,USERCODE = <usercode> -----|
```

Error Messages

INVALID REMOTE HOST

- if <your-host> is specified and not valid (either because HOST VALIDATE is true and the host is not in the valid list, or because the host is unreachable).

NO SUCH HOST

NO SUCH USERCODE

USERCODE MAY NOT BE SPECIFIED

- if GUARDED or PUBLIC security applies.

DUPLICATE AGENT NAME

SECTION 4

ATTRIBUTE ENTRY COMMANDS

Attribute Entry commands set single-valued attributes in the Network Services managers. The syntax for these commands include both the Attribute Entry and the Inquiry forms:

```
-- <attribute-name> ----->|
      | -- <set-attribute-specification> - |
```

When the attribute name alone is entered, it is an Inquiry command; otherwise it is an Attribute Entry Command.

Empty strings can be entered explicitly by using *NULL in place of <value>. Empty strings are displayed in responses and reports in this way.

If *DEFAULT is used in place of <value>, the default value is selected. This is particularly useful after initialization to return the attribute to the default value.

LOCAL IDENTITY COMMAND

Command Syntax

```
LOCALIDENTITY ----->|
---
      | --- #<node-address> ----- |
      | --- <host-name> #<node-address> --- |
```

<node-address> range: 1 to MAX NODE ADDRESS
or less than 65534

Semantics and Mode Serviced

If <node-address> is specified, this command sets the node address of the local node. If <host-name> is specified, the host name of the local node is compared with this host name (entered by HN command), and a negative response is returned if they are different.

The <node-address> is required to be entered during the Network mode Attribute Entry phase; it cannot be entered at any other time.

Default Value

There is no default value. This MUST BE ENTERED.

Examples

```
LOCALIDENTITY = #12  
LOC = CENTER #12
```

Response Syntax

```
LOCALIDENTITY = <host-name> -----|  
                    | - #<node-address> - |
```

NOTE

The node address is omitted if it has not yet been set during the attribute entry phase.

Error Messages

INVALID NODE ADDRESS

- if the <node-address> value exceeds the range 1 to 65534, or is greater than MAX NODE ADDRESS, if one is specified.

HOST NAME DOES NOT MATCH

- if <host-name> is entered and it does not match the current <host-name> value.

GET-ONLY ATTRIBUTE COMMANDS

Get-only attributes are not settable through the Operations Interface. Some of them are computed (based on other attribute values), or set externally to Network Services operations.

As these commands are inquiries only, the format is simplified to the command syntax followed by the response and any comments.

PORTLEVELINCARNATION -----|
- - - - -

This command displays the current value for the Port Level Incarnation attribute.

PORTLEVELINCARNATION is updated at every initialization of the Port Level and every time Network mode is entered.

Response Syntax

PORTLEVELINCARNATION = <string>

ROUTERHEADERSIZE -----|
- - - - -

This displays the size of the Router header in bytes.

Response Syntax

ROUTERHEADERSIZE = <integer> (bytes)

TRANSITCOUNTLIMIT -----|
- - - - -

This displays the maximum value allowed for the Router frame field, TCNT (transit count).

Response Syntax

TRANSITCOUNTLIMIT = <integer>

VERSION -----

This displays the version identifications for the various BNA Network Services levels.

Response Syntax

```
VERSION FOR |<-----, -----|
             |-- NETWORK ---- = <string> -----|
             | - NSM ----- = <string> -----|
             | - PORTLEVEL -- = (<level-version>) -|
             | - ROUTER ---- = (<level-version>) -|
             | - STATIONLEVEL = (<level-version>) -|
```

where <level-version> is:

```
             |<-----, -----|
-- <string> : <integer> COMPATIBLE WITH --- <integer> ---|
notes:
```

- . The <string> is the level version identifier.
- . The <integer>s are the level version protocols.
- . In Isolated mode, only Network Services Manager (NSM) and Port Level Manager (PLM) versions are returned.

NETWORK ORIENTED COMMANDS

Command Syntax

```
NETWORKMAXSEGMENTSIZ -----|  
- - - - - | - = <integer> -- |
```

<integer> range: 60 to 65535

```
NETWORKVERSION ----->|  
- - - - - | - = <string> -- |
```

<string> range: 1 to 17 significant characters

Semantics

These commands set the NETWORK VERSION and the NETWORK MAX SEGMENT SIZE. The NETWORK VERSION can be used by the network administration to coordinate changes in network configurations; the NETWORK VERSION must be identical for all communicating nodes.

The NETWORK MAX SEGMENT SIZE is the MAX SEGMENT SIZE used for all communications in the network (with the possible exception of link connections between two consenting neighbors).

The segment size is the number of bytes used by a Router frame. This is the Router header plus any port frame characters. It is the size of the message that is sent from a node, excluding any Station Level header or trailer characters.

Servicing Phase

These can be entered only during the Attribute Entry phase.

**NETWORKMAXSEGMENTSIZ
NETWORKVERSION**

Default Value

These commands are optional, with default values:

NETWORK VERSION = *NULL (the empty string)
NETWORK MAX SEGMENT SIZE = 128 bytes

Examples

NETWORKVERSION = BNA-1A
NWSS = 256

Response Syntax

NETWORKVERSION = <string>
NETWORKMAXSEGMENTSIZ = <integer>

Error Messages

INVALID NETWORK MAX SEGMENT SIZE
- if <integer> exceeds allowable range of values.

Default Value

This command is optional; the default value is STANDARD.

Examples

```
LOGGING + MINIMUM
LOG +
```

Response Syntax

```
LOGGING = -----|
          |-----+-----|
          |             |
          | - MINIMUM ---|
          | - MAXIMUM ---|
```

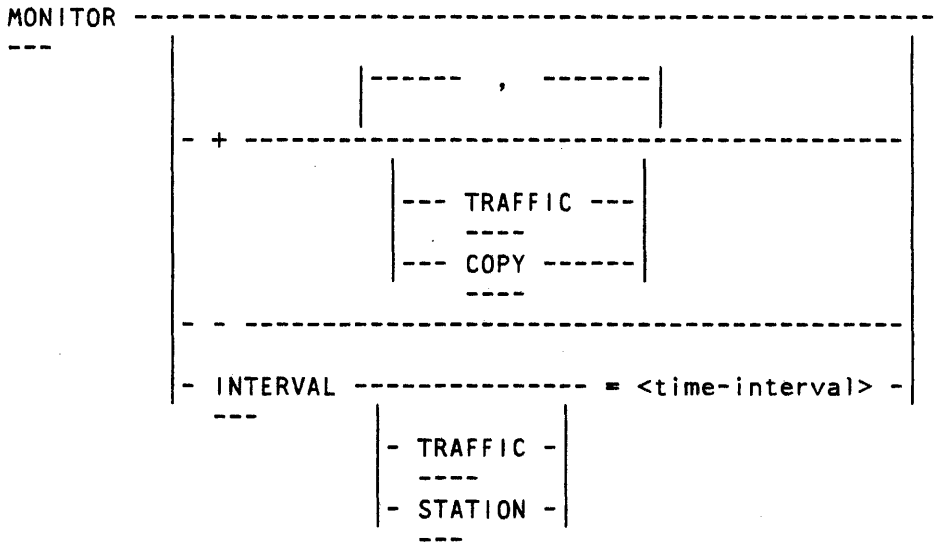
where + alone indicates standard logging,
and - indicates no logging.

Error Messages

```
INVALID LOGGING OPTION
- if option specified is other than MINIMUM or MAXIMUM.
```


MONITOR TRAFFIC COMMAND

Command Syntax



<time-interval> range: 1 to 60 mins.

Semantics

This command sets the monitor or monitor interval for selected activities, and supercedes any previous such command.

MONITOR + or - form:

If the monitor is turned on (MONITOR +) and TRAFFIC ONLY is specified, a summary of all traffic passing through the Router is recorded in the log. If COPY ONLY is specified, a copy of each Router frame is logged. If MONITOR + is unqualified, both items are recorded in the log.

The command MONITOR - resets the monitor for both the TRAFFIC and COPY options.

This command cannot turn on or turn off the Station Monitors. These monitors are accessed by setting a station MONITOR attribute using the ADD STATION or MODIFY STATION command.

MONITOR INTERVAL form:

The MONITOR INTERVAL form of the command sets the monitor time interval, either for Router traffic (TRAFFIC specified), or for station monitors (STATION specified), or for both (the unqualified MONITOR INTERVAL form).

This time interval is used to trigger the recording of monitor information in the Router, if the TRAFFIC monitor has been turned on, or in any station where the monitor attribute has been set to TRUE.

Servicing Phase

These commands can be entered at any time while in Network mode.

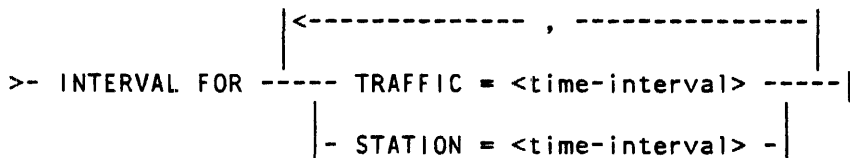
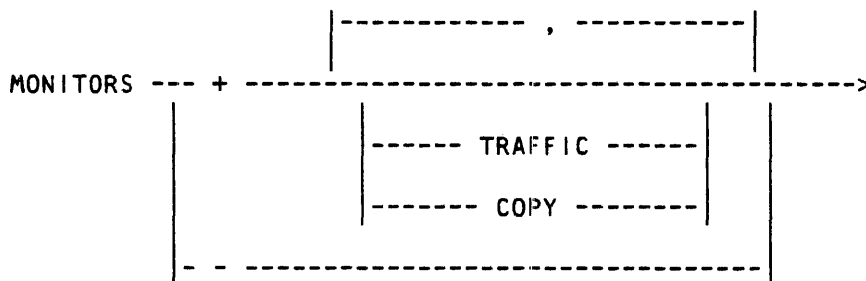
Default Value

This command is optional. The default value for the MONITOR is OFF, and the default value for INTERVAL is 15 minutes.

Examples

```
MON +
MONITOR + TRAFFICONLY
MON -
MONITOR INTERVAL = 45:0
MON INT STA = 30:0
```

Response Syntax



Error Messages

INVALID MONITOR OPTION

- if other than the specified options are entered with this command.

INVALID MONITOR TIME INTERVAL

- if time interval is outside allowable range.

PORT LOG INTERVAL COMMAND

Command Syntax

```
PORTLOGINTERVAL -----|  
- - - | - = --- <time-interval> --|  
      <time-interval> range: 1 to 1440 minutes (1 day)
```

Semantics

This command sets the time interval which starts the recording of certain resource usage information in the Port Level.

Servicing Phase

This command can be entered at any time.

Default Value

This command is optional. The default value is 60 minutes.

Examples

```
PORTLOGINTERVAL = 120:0  
PLI = 6:0:0
```

Response Syntax

```
PORTLOGINTERVAL = -- <time-interval> -----|
```

Error Messages

INVALID PORT LOG TIME INTERVAL
- if time interval is outside allowable range.

TUNING ATTRIBUTE COMMANDS

Tuning commands are used to optimize network performance by adjusting attributes in the Network Services levels: Port, Router, and Station.

Although all of these tuning attributes are accessible to a network Operations Interface agent, they should not be adjusted without regard to their effects. In particular, the attributes PORTSEGMENTTIMEOUT and PORTWINDOWSIZE have calculated default values, and rely upon MAXRESISTANCEFACTOR and MAXHOPCOUNT as input to the calculation. Careful consideration of the effects of changing these should be made, as indiscriminate changes can adversely affect network performance.

In the following paragraphs the attributes are described, and where caution is needed, notes and guidelines are provided.

NOTE

The format for this subsection is slightly different. The commands are all mentioned as a group and are then followed by responses, defaults, errors, and so on.

ROUTING ORIENTED COMMANDS

These commands set the routing-oriented tuning attributes.

Command Syntax

```
MAXHOPCOUNT -----|  
- - - | - = --- <integer> -----|  
      |  
      | <integer> range: 1 to 255
```

See MAXRESISTANCEFACTOR for explanation.

```
MAXRESISTANCEFACTOR -----|  
- - - | - = --- <integer> -----|  
      |  
      | <integer> range: 1 to 65535
```

The attributes MAXRESISTANCEFACTOR and MAXHOPCOUNT specify the bounds of the reachable nodes. That is, the BIAS[®] mechanism considers each path to a node, and limits its search when either of these attribute values is reached. Paths to nodes that equal or exceed these limits are considered unusable, and nodes that are only accessible by paths that equal or exceed these limits are unreachable.

The hopcount should be kept as small as possible but still allow all reasonable routings.

The maximum resistance should be made large enough to allow all reasonable routings.

NOTE

Both MAX HOP COUNT and MAX RESISTANCE FACTOR values must be made 1 more than the maximum values calculated for resistance and the number of hops to be used across the network.

For further detail on the selection of these values, refer to Appendix B.

```
NEIGHBORRESTARTTIMEOUT -----|
- - - - - | - = --- <time-interval> -- |
```

<time-interval> range: 0 to 1200 tenths-of-seconds
(2 minutes).

This timeout value is used whenever a NEIGHBOR-RESTART command is received. While this timer runs, the Router does not send NETCHANGES about its neighbor.

This attribute specifies the time that a node waits, after a neighbor has been recognized as restarting, before it informs other nodes of the restarting neighbor's view of the network. The default value should not need adjustment.

```
NODERESISTANCEFACTOR -----|
- - - - - | - = --- <integer> ----- |
```

<integer> range: 1 to 65535

This is the resistance factor of the local node, and is the average time, in milliseconds, for a message to transit the node in the absence of any other traffic. It is used by other nodes for choosing routings.

This attribute can be increased to make it a less desirable transit path by other nodes. It can be decreased to make it more desirable.

[®] BIAS is a trademark of Burroughs Corporation.

```

NODEUPTIMEOUT -----|
-   -   -           | - = --- <time-interval> --|

```

<time-interval> range 0 to 600 tenths-of-seconds
(1 minute)

When a node first comes up, no NETCHANGE messages are sent out until a timeout period of NODEUPTIMEOUT occurs with no LINKCHANGE or NETCHANGE messages received.

This attribute is used to prevent an unnecessary number of netchange messages from being sent by this node after it enters the network, either initially or during recovery. The node should wait until all netchange messages are received (and analyzed) before sending netchanges to its neighbors.

The default time value should not require adjustment.

Servicing Phase

The NODE RESISTANCE FACTOR, NODE UP TIMEOUT, and NEIGHBOR RESTART TIMEOUT attributes can be set at any time while in Network mode. The others can be entered only during the Network mode Attribute Entry phase.

Default Value

```

MAX RESISTANCE FACTOR    = 5000 (these are arbitrary values, they
MAX HOP COUNT            = 5      should be changed to reflect the
                               network needs)

```

```

NODE UP TIMEOUT          = 5.0 (seconds)
NEIGHBOR RESTART TIMEOUT = 10.0 (seconds)

```

NODE RESISTANCE FACTOR varies with system type:

```

B1000 type system = 150
B4800 type system = 70
B6800 type system = 30
B7800 type system = 10

```

Examples

```

MRF = 7500
NODEUPTIMEOUT = 7.5

```

Response Syntax

```
|--- <attribute-name> -- = --- <integer> -----|  
| - <time-interval> -- |
```

for example, NEIGHBORRESTARTTIMEOUT = 1:5.5

Error Messages

INVALID MAX RESISTANCE FACTOR

- if integer is outside of the allowable range.

INVALID MAX HOP COUNT

- if integer is outside of the allowable range.

INVALID NODE RESISTANCE FACTOR

- if integer is outside of the allowable range.

INVALID NODE UP TIMEOUT VALUE

- if time interval is outside of the allowable range.

INVALID NEIGHBOR RESTART TIMEOUT VALUE

- if time interval is outside of the allowable range.

LINK RESISTANCE TO NEIGHBOR COMMAND

```
LINKRESISTANCE --- <neighbor> = --- <integer> -----|
- - - - - | -- *DEFAULT -- |
```

<integer> range: 1 to MAXRESISTANCEFACTOR

Semantics

This command sets the link resistance factor for the link to the specified neighbor in the Router Neighbor Table. It is the average time, in milliseconds, for a message to transit the link.

If this value is set to an integer greater than zero, it overrides the calculated default value of link resistance.

This value is used by the Router in choosing routings. The default value should not need adjustment except for altering the desirability of using a link. For example, there may be a case where the link to a neighbor is required for communication to that neighbor, but it is not required to be used for traffic to transit this node. In this case, if the link resistance is increased, it still exists for the neighbor, but other nodes will find that the total link resistance is higher than MAXRESISTANCEFACTOR, or higher than an alternate path. Thus, only the neighbor will use the link.

Default Value

If the value is *DEFAULT, the default value is calculated by the following equation:

$$\text{default link resistance} = \frac{1000 \times \text{maxsegsz} \times 8}{\text{summation}(\text{speed} \times \text{eff}/100)}$$

NOTE

Where speed and eff are the Station Level attributes SPEED and EFFICIENCY for the individual links. The summation is used where there are multiple-parallel links.

Examples

```
LINKRESISTANCE REMOTE-1 = 2000  
LR REMOTE-2 = 4567
```

Response Syntax

```
LINK RESISTANCE CHANGED WITH <neighbor>
```

Error Messages

```
INVALID LINK RESISTANCE FACTOR  
- if integer is outside allowable range.
```

SUPPORT COMMUNICATION COMMANDS

Support Communication commands are used to set tuning attribute values for support communication. Changing these attributes only affect new supports, existing support attributes remain unaffected.

Command Syntax

```
PORTCOMPRESSIONALLOWED -----|  
- - - - - |  
| --- |  
| - + - |
```

When set, +, this allows the supports to use data compression. Compression is only invoked if both supports request it.

PORTRESUMEREADY
PORTRETRYLIMIT
PORTSEGMENTTIMEOUT
PORTWINDOWSIZE

```
PORTRESUMEREADY -----|
- - - - - | - = --- <integer> -----|
```

<integer> range: 1 to 99 (per cent)

PORTRESUMEREADY regulates the level at which the MESSAGE FORWARDING QUEUE changes from NOT READY to READY; that is, the queue becomes not ready when full but only goes ready again when PORTRESUMEREADY queue level is reached. This is used to reduce unnecessary changes of ready status in those subports communicating with "remote" subports.

```
PORTRETRYLIMIT -----|
- - - - - | - = --- <integer> -----|
```

<integer> range: 0 to 100

This attribute specifies the number of times to retransmit any particular unacknowledged subport segment before disconnecting the dialogue.

```
PORTSEGMENTTIMEOUT -----|
- - - - - | - = --- <time-interval> --|
```

<time-interval> range: 0 to 65535 secs.

This subport attribute specifies the waiting time for acknowledgement of any particular subport segment before attempting to retransmit it or report an error.

The segment timeout is intended to recognize that subport frames are lost, probably due to an intransit node failure, or closure. This attribute has a calculated default value and SHOULD NOT be changed without CAREFUL CONSIDERATION of the effect. If changed, the value should NOT be made too SMALL. If it is made too small it can cause excessive retransmissions in the network when the network load is at a peak, and transit times are at a maximum.

```
PORTWINDOWSIZE -----|
- - - - - | - = --- <integer> -----|
```

<integer> range: 1 to 60

This attribute indicates the number of segments that subports can send (and the number it can expect to receive) that are unacknowledged. The value applies to traffic in both directions. The actual value used for a dialogue is determined during subport attachment and if they are unequal, the smallest value of PORTWINDOWSIZE of the two ports is used.

Section 4

Attribute Entry Commands

Tuning Attribute Commands

PORTWINDOWSIZE has a calculated default value and SHOULD NOT be changed without CAREFUL CONSIDERATION to the effect. If changed, the value should NOT be made too LARGE, as this can lead to overload of the network. The network peak load is, in fact, a function of the summation of all window sizes used within the network. Exceeding the peak load capability of any part of the network can cause extra retransmissions, thus degrading the network performance.

Servicing Phase

These commands can be entered only in Network mode.

Default Value

The following attributes, if not explicitly set during the Attribute Entry phase of node initialization, have default values computed as follows:

PORT WINDOW SIZE = 4 x MAX HOP COUNT

PORT SEGMENT TIMEOUT = PORT WINDOW SIZE X MAX RESISTANCE FACTOR
divided by 1000 and rounded up to an integer

The values of MAX HOP COUNT and MAX RESISTANCE FACTOR used in the calculations are the values which exist at the end of the node initialization.

These attributes have the following default values; the set commands are optional:

PORT RETRY LIMIT = 2
PORT RESUME READY = 60 (percent)
PORT COMPRESSION ALLOWED = - (off)

Examples

PORTSEGMENTTIMEOUT = 10
PWS = 5

Response Syntax

```
|-- <attribute-name> --- = --- <integer> -----|
|
| - <time-interval> -
|----- + -----
|----- - -----
|----- *UNKNOWN -----|
```

For example: PORTWINDOWSIZE = 5

NOTE

Inquiries about values of calculated attributes return *UNKNOWN if they have not yet been determined.

Error Messages

- INVALID PORT SEGMENT TIMEOUT INTERVAL
- if time interval is outside allowable range.
- INVALID PORT WINDOW SIZE
- if integer is outside of allowable range.
- INVALID PORT RETRY LIMIT
- if integer is outside of allowable range.
- INVALID PORT RESUME READY FACTOR
- if integer is outside of allowable range.
- INVALID PORT COMPRESSION ALLOWED OPTION
- if value is other than + or -.

INTER-HOST COMMUNICATION COMMANDS

The following commands set the tuning attributes used in host-to-host communications control.

Command Syntax

```
HOSTINACTIVEDISCONNECT -----|  
- - - - - |  
            |-----+-----|  
            |----- - -----|
```

This attribute, if + (true), indicates that a PLM-to-PLM dialogue is to be terminated automatically if, for a continuous time interval determined by the HOST INACTIVE TIMEOUT attribute, there is no support allocated for use with the associated remote host on behalf of any using process.

```
HOSTINACTIVETIMEOUT -----|  
- - - - - |  
            | - = --- <time-interval> -- |
```

<time-interval> range 0 to 1440 minutes (1 day)

This attribute is the amount of time a PLM-to-PLM dialogue is maintained without any supports allocated for use with a given host (this does not include the PLM-to-PLM supports). It is only meaningful when the HOST INACTIVE DISCONNECT is + (true).

```
HOSTUNREACHABLETIMEOUT -----|  
- - - - - | - = <time-interval> ----|
```

<time-interval> range 0 to 1440 minutes (1 day)

This sets the time period that the local host waits while a remote host is unreachable before terminating all port level communications with that host.

Servicing Phase

These commands can be entered at any time while in Network mode.

Default Value

These commands are optional. The default values are:

```
HOST INACTIVE DISCONNECT = - (false)  
HOST INACTIVE TIMEOUT = 10:0 (10 minutes)  
HOST UNREACHABLE TIMEOUT = 30:0 (30 minutes)
```

Examples

```
HOSTINACTIVEDISCONNECT = 20:0  
HID = -
```

Response Syntax

```
|--- <attribute-name> --- = --- <integer> -----|
|
| - <time-interval> -
|----- + -----
|----- - -----
|----- *UNKNOWN -----|
```

NOTE

Inquiries about values of calculated attributes return *UNKNOWN if they have not yet been determined.

Response Syntax Examples

```
HOSTINACTIVEDISCONNECT = +
HOSTUNREACHABLETIMEOUT = 10:0
```

Error Messages

INVALID HOST INACTIVE DISCONNECT OPTION
- if option is other than TRUE/FALSE.

INVALID HOST INACTIVE TIMEOUT
- if time interval is outside allowable range.

INVALID HOST UNREACHABLE TIMEOUT
- if time interval is outside allowable range.

STATION-ORIENTED COMMANDS

Command Syntax

```
NEIGHBORBUSYTIMEOUT -----|  
- - - - - |-----|  
          |-----<time-interval>-----|
```

<time-interval> range: 0 to 3000 tenths of seconds
(5 mins)

This command sets the time period that ALL links to a neighbor must be busy before a NEIGHBOR STATUS REPORT is generated. Neighbors are only reported as being busy if the station attribute REMOTEBUSYMONITOR is set.

```
NEIGHBORGREETINGTIMEOUT -----|  
- - - - - |-----|  
          |-----<time-interval>-----|
```

<time-interval> range: 0 to 600 tenths-of-seconds
(1 minute)

This command sets the timeout value for the Station Level Greetings. It is used to set a maximum time period within which the validation process must be accomplished.

Servicing Phase

These commands can be entered at any time during Network Mode.

Default Value

```
NEIGHBOR GREETING TIMEOUT = 30.0 seconds.  
NEIGHBOR BUSY TIMEOUT    = 60.0 seconds.
```

Both attributes are optional, and default if not explicitly set.

Examples

```
NEIGHBORGREETINGTIMEOUT = 45  
NGTO = 27.5
```

```
NEIGHBORBUSYTIMEOUT = 2:0.0  
NBTO = 2000
```


Response Syntax

NEIGHBORGREETINGTIMEOUT = <time-interval>

NEIGHBORBUSYTIMEOUT = <time-interval>

Error Messages

INVALID NEIGHBOR GREETING TIMEOUT VALUE
- if time interval is outside of the allowable range.

INVALID NEIGHBOR BUSY TIMEOUT VALUE
- if time interval is outside of the allowable range.

TUNINGS INQUIRY COMMAND

The Tunings inquiry command inquires of the state of all tuning attributes for a node, and responds with a list of all of the attribute settings.

```
TUNINGS -----|  
-----
```

Response Syntax

All of the TUNING attributes (described in this subsection) are listed together with their values or settings. Attributes included are:

HOST INACTIVE DISCONNECT, HOST INACTIVE TIMEOUT
HOST UNREACHABLE TIMEOUT, LINK RESISTANCE
MAX HOP COUNT, MAX RESISTANCE FACTOR
NEIGHBOR BUSY TIMEOUT, NEIGHBOR RESTART TIMEOUT
NEIGHBOR GREETING TIMEOUT
NODE RESISTANCE FACTOR, NODE UP TIMEOUT
PORT COMPRESSION ALLOWED, PORT RESUME READY
PORT RETRY LIMIT, PORT SEGMENT TIMEOUT
PORT WINDOW SIZE

ATTRIBUTE SUMMARY

All of the attribute settings and values for the node may be obtained with the NODESUMMARY inquiry.

```
NODESUMMARY -----|  
-----
```

Response Syntax

All attributes and their settings or values are displayed as a list.

Servicing Phase

This can only be entered in Network mode.

NOTE

The Candidate Summary and the Mode information are also applicable in Isolated mode, but these must be explicitly requested using the CANDIDATE and NET inquiry commands, respectively.

SECTION 5

CONFIGURATIONAL COMMANDS

The configurational commands generate and change the node's network configuration information. They are operationally inert; that is, they neither create nor destroy communication paths. They simply change the potential for what paths can be created.

For example, ADD STATION or ADD CONNECTION does not create a new link to a neighbor; rather it sets up the information by which a link can be later created with an operational command such as ESTABLISH CALL. Similarly, DELETE STATION does not sever a link; the station must already be disconnected in order to be deleted.

In the DELETE commands, if both <host-name> and <node-address> are specified, the correspondence of the two is checked before the delete takes place.

Unless otherwise specified, all configurational commands can be entered only during the Initialization or Operating phases of Network mode. The first Configurational command entered during node initialization signals the end of the Attribute Entry phase. All required attributes must have been entered prior to this point.

In the examples, the symbol => indicates that the positive response to the given command follows.

CONFIGURATIONAL COMMAND SYMMETRY

Consider the ADD and DELETE commands for hosts and nodes and the ADD and DELETE STATION or CONNECTION commands to be described later. The action of these commands possesses a symmetry which is illustrated in figure 5-1. In this diagram "-" implies no effect.

	Port-level Effect -----	Router Effect -----	Station-level Effect -----
ADD HOST	add host	add node	- - -
DELETE HOST	delete host	- - -	- - -
ADD NODE	- - -	add node	- - -
DELETE NODE	delete host	delete node	(must not be neighbor)
ADD CONNECTION - - - WITH NEIGHBOR		(must already be a node if Router VAL true)	add connection
DELETE CONN. - - - WITH NEIGHBOR		- - -	delete connection

Figure 5-1. Configurational Command Symmetry

NETWORK DEFINITION COMMANDS

ADD HOST COMMAND

Command Syntax

```
ADD HOST <host-name> #<node-address> -----|  
--- ----
```

Response Syntax

```
<host-name> #<node-addr> ADDED AS HOST
```

Semantics

This command adds the specified host as a remote host. When HOST VALIDATE is true, entering this command is required to validate the host host valid. Otherwise the only effect is to add the host for the purpose of translating between <host-name> and <node-address>.

If ROUTER VALIDATE is true (the VALIDATE option was set to ALL), this command adds the host as a node to the Router unless the node already exists (by having entered previously an ADD NODE command).

Examples

```
ADD HOST BRANCH2 #5  
  
=> BRANCH2 #5 ADDED AS HOST
```

Error Messages

```
HOST NAME - NODE ADDRESS DO NOT MATCH  
- if a previous command giving both host-name and node-address was entered  
and this command has a different host-name and node-address pair.
```

```
INVALID NODE ADDRESS  
- if the <node-address> value exceeds the range 1 to 65534, or is greater  
than MAX NODE ADDRESS, if one is specified.
```

ADD NODE COMMAND

Command Syntax

```
ADD NODE ----- #<node-address> -----|
--- ---- | - <host-name> - |
```

Response Syntax

```
----- #<node-addr> ADDED AS NODE -----|
| - <host-name> - |
```

Semantics

If ROUTER VALIDATE is true, this command adds the specified node to the Router existence. If, in addition the host name is supplied, the host-name and node-address pair information is stored for translation purposes, whether or not ROUTER VALIDATE is true.

Examples

```
ADD NODE BRANCH2 #5
=> BRANCH2 #5 ADDED AS NODE
```

Error Messages

ROUTER VALIDATE IS FALSE, COMMAND HAS NO MEANING

- if this command specifies only node address and ROUTER VAL is false.

NODE ALREADY ADDED

- if node was already added and this command does not specify host-name.

HOST ALREADY ADDED

- if host validate is true and host is in the table.

MISSING NODE ADDRESS

INVALID NODE ADDRESS

- if the <node-address> value exceeds the range 1 to 65534, or is greater than MAX NODE ADDRESS, if one is specified.

DELETE HOST COMMAND

Command Syntax

```
DELETE HOST <host-name> -----|
---  ---- |
          | -- #<node-address> -- |
          |
          | <node-address> range: 1 to 65534
          | (or MAX NODE ADDRESS if specified)
```

Response Syntax

```
<host-name> #<node-address> DELETED AS HOST
```

Semantics

This command has meaning only if HOST VALIDATE is true, in which case the indicated host is removed as a valid remote host. Whether HOST VALIDATE is true or not, the host-name and node-address pairing information is retained for translation purposes.

A PLM-to-PLM support to a host that is to be deleted is allowed. However, the host is not deleted and an error is returned if there is at least one other support in use to that host.

Examples

```
DEL HOST BRANCH2
=> BRANCH2 #5 DELETED AS HOST
```

Error Messages

```
HOST NOT DELETED, SUBPORTS IN USE: <integer>
```

```
HOST NAME - NODE ADDRESS DO NOT MATCH
- if this command supplies both and they do not agree with the existing
  host-name and node-address pair information.
```

```
INVALID HOST NAME
- if host name = local host.
```

INVALID NODE ADDRESS

- if the <node-address> value exceeds the range 1 to 65534, or is greater than MAX NODE ADDRESS, if one is specified.

DELETE NODE COMMAND

Command Syntax

```
DELETE NODE ----- #<node-address> -----|
---  ---  | -- <host-name> -- |
                <node-address> range: 1 to 65534
                (or MAX NODE ADDRESS if specified)
```

Response Syntax

```
----- #<node-address> DELETED AS NODE -----|
| - <host-name> - |
```

Semantics

This command performs the following functions: (1) it deletes the specified node from the Router existence, (2) it deletes the specified node as a host.

If the Host Validate is false, and the node is still reachable, the PLM (Port Level Manager) will attempt to send greetings to establish the identity of the remote host. If successful, then the new identity is used for translation purposes, else no translation of host/node is possible.

The node is not deleted and an error is returned if the node is connected as a neighbor, or there is at least one subport, not including the PLM subports, in use to that node.

Examples

DELETE NODE #5

=> BRANCH2 #5 DELETED AS NODE

Error Messages

NODE NOT DELETED, NODE IS CONNECTED AS NEIGHBOR

NODE NOT DELETED, SUBPORTS IN USE: <integer>

HOST NAME - NODE ADDRESS DO NOT MATCH

- if both are entered and they do not agree with pair information previously entered.

NODE ALREADY DELETED

- if node does not exist or ROUTER VALIDATE is false, and the node does not appear as a remote host.

INVALID NODE ADDRESS

INVALID HOST NAME

STATION LEVEL COMMANDS

These commands enter configurational information for the Station Level. They mainly set up the station names and the associated station attributes. In this subsection the station attributes are first described followed by the Station Level commands.

STATION ATTRIBUTES

Station attributes are set using ADD and MODIFY STATION commands. Also, these attributes are used by the ADD PROFILE command to construct a named list of attributes for later use by an ADD STATION. The attribute descriptions are organized under several categories, and within each category they are in alphabetic order. At the end of the attribute descriptions there is a list, figure 5-2, of BDLC Station Level attributes, showing default values and the use of the attribute.

Station-Level-Manager Attributes

AUTOINIT = TRUE or FALSE optional, default = TRUE

For certain station types AUTOINIT TRUE causes the station to be attached, at initialization, to its neighbor station without an explicit Operations Interface command.

EFFICIENCY = <integer> optional, default = 95% for DEDICATED station
--- = 45% for SWITCHED station

<integer> range: 1 to 100%

This is used for determining the default value for the link resistance factor as used by the Router. It is used in conjunction with line speed for this determination. Efficiency is entered as a percentage (%).

HARDWARE = <string> required

It is used to identify the hardware to be used by a station. This is a system-dependent attribute; for this, refer to the publication for the specific product.

MAX SEGMENT SIZE = <integer> optional, default = NETWORKMAXSEGMENTSIZ
--- - -

<integer> range: 60 to 65535 bytes

The MAX SEGMENT SIZE is the largest link information unit that can be sent by the Router to the Station Level. For BDLC links, it is the length of the l-Field.

This attribute must be set at least as large as NETWORKMAXSEGMENTSIZ. For Port dialogues that are between hosts that are neighbors, the station MAXSEGMENTSIZ is used. For port dialogues that are between hosts that are not neighbors, the NETWORKMAXSEGMENTSIZ is used.

MONITOR = ON or OFF optional, default = OFF
--- -- ---

The setting of this attribute causes the station to accumulate count information that is periodically interrogated and logged (refer to MONITOR command).

SPEED = <integer> required

This is the line speed (Baud rate). It is used to determine the default link resistance factor, as used by the Router.

Optional Station Attributes

DATARATESELECTOR = ON or OFF default = OFF
- - --- -- ---

This attribute causes the data rate selector level to a modem to be high ON, or low OFF.

RETRYLIMITCHECKPT = <integer> default = 3
- - - -

This attribute sets the retry limit for attempts to obtain checkpoint information. This can occur when the P-bit is put on to obtain acknowledgement of previous transmissions. If there is no response after the retry limit is reached, the BDLC station attempts a link reset (sends SABM).

**STATION ATTRIBUTES:
RETRYLIMITSABMDIS**

RETRYLIMITSABMDIS = <integer> default = 3
- - -----

This attribute sets the retry limit for unacknowledged SABM or DISCONNECT transmissions for the BDLC station. SABM retry limit being exceeded causes the station (not permanent type) to attempt a disconnect. If this fails the station is closed and the Station Level manager informed.

SELECTSTANDBY = ON or OFF default = OFF
- - -- ---

This attribute causes the select standby level to a modem to be high ON, or low OFF.

WINDOWSIZE = <integer> default = 7
- -

<integer> range: 1 to 7

This attribute is used to determine the number of BDLC l-frames that can be sent before an acknowledgement is received.

Required Station Attributes

The following attributes are required for all stations.

CTSRPTIMER = <time-interval>

range: 8.5 to 275 milliseconds

This attribute describes the time to wait between request-to-send (RTS) to receiving clear-to-send (CTS) from the modem.

DCDOFFDELAYTIMER = <time-interval>

range: 1 to 60 secs.

This attribute describes the time to wait during transitions of DCD (data-carrier-detect) from the modem. This is set to avoid unnecessarily closing the link due to a short hit on the line.

DIALRESPONSETIMER = <time interval>
- - - - -

range: 3.5 to 40.5 secs.

This attribute is set to allow the system to disconnect the line if for any reason an incoming call does not result in coherent dialogue. This can occur if a human dials into the system by mistake or otherwise.

DSRRSPTIMER = <time-interval>

This attribute describes the time to wait for the data set ready (DSR) modem level to come true. This occurs in several instances, for example, when data terminal ready (DTR) goes true and all other conditions are right, DSR should go true after some modem-defined time. If this does not occur, then the station is closed.

FRSTIMERSHORT = <time-interval>
---- -

This attribute, 'F' Response Timer Short, is the time allowed to receive an 'F' bit response after a 'P' bit has been sent out. If the timer expires, the command is retransmitted.

This timer and the FRSPTIMERLONG attribute are used for the same functions. However, the two values are provided to overcome contention situations of two-way-alternate operation. (The timer chosen for the actual function performed depends upon the factors of node addresses, and whether it is an input or output operation, and is handled internally by the station).

The timer value should be set long enough that normal communication procedures will not cause the timer to time out. Factors to be considered for arriving at a value should be: propagation delays (in both directions); any modem turn-around delays; the command message and I-Frame message transmission times (functions of message length and line speed); and the end-point processing delays.

FRSPTIMERLONG = <time-interval>

This attribute, 'F' Response Timer Long, has the same function as FRSPTIMERSHORT, but should be set 1 second longer. See FRSPTIMERSHORT for detail.

STATION ATTRIBUTES:
CHECKPOINTTIMER

CHECKPOINTTIMER = <time interval>
- - -

This attribute describes the time to wait after an I-frame is sent before demanding an acknowledgement. This timer is restarted upon each receipt of an I-frame until the last outstanding I-frame is received.

This should be set to a value at least twice as large as FRSPTIMERSHORT. (As this timer is used to monitor I-frame transmissions, the time for frame transmission may be significant and should not be overlooked.)

Optional Station Event-Monitor Attribute

CPDMONITOR = ON or OFF optional, default = OFF

Setting this attribute ON will cause instances of open or close Connection Port Dialog to be logged.

Optional Station Counter-Monitor Attributes

The following attributes are counters that can be accessed for statistical information about the station. They are all reset initially when the station is opened and are never reset. They do wrap-around at their maximum values. (Refer to publications for specific system product.) There is no indication of a wrap-around, thus it is necessary to interrogate them frequently enough to be able to recognize the condition.

FRAMERECEIVEDCOUNTER
- - -

This attribute contains the number of frames received by the station.

FCSFAILURECOUNTER
- - -

This attribute contains the number of FCS failures recorded by the station.

FRAMESENTCOUNTER
- - -

This attribute contains the number of frames sent by the station.

IFRAMESENTCOUNTER

-- - -
This attribute contains the number of I-Frames sent by the station.

REMOTEBUSYMONITOR = SET or RESET

- - - -
This attribute should be set to allow the station to inform the Station Level manager when a neighbor becomes busy. This is used in conjunction with the NEIGHBOR BUSY TIMEOUT. This is used to allow logging of the neighbor link busy condition.

ACU-Station Attributes

The following are required for switched BDLC stations with ACU only:

ACURSPTIMER = <time-interval>

range: 8.5 to 40.5 secs.

This timer attribute sets the time interval between ACU (automatic call unit) logical steps, for example, between call-request and data-line-occupied, or between digit-present and present-next-digit. If this timer times out, it would indicate a failure to make the outgoing call. Local telephone company requirements should be consulted for this value.

DIALRETRYCOUNT = <integer>

- - - -
This attribute sets the limit of attempts to make a physical connection using auto-dial facilities. This limit is usually specified by the local telephone company, so refer to their requirements.

DPRDELAYTIMER = <time-interval>

range: 2 to 10 milliseconds

This timer attribute sets the timer that determines the interval between the digits being placed on the modem line and the time that the modem is informed of their presence by digit-present level to the modem. Local telephone company requirements should be consulted for this value.

**STATION ATTRIBUTES:
RETRYDELAYTIMER**

RETRYDELAYTIMER = <time-interval>

range: 1 to 5 mins.

When making auto-dial calls this timer attribute sets the time between the end of one call, or call attempt and the next. This time is a local telephone company requirement.

ATTRIBUTE NAME	default	required or optional	For use by
ACURSPTIMER		required	ACU - attribute
AUTOINIT	true	optional	Station Level Manager
CTRSPTIMER		required	BDLC Station Group
DATARATESELECTOR	off	optional	BDLC Station Group
DCDOFFDELAYTIMER		required	BDLC Station Group
DIALRESPONSETIMER		required	BDLC Station Group
DIALRETRYCOUNT		required	ACU - attribute
DPRDELAYTIMER		required	ACU - attribute
DSRRSPTIMER		required	BDLC Station Group
EFFICIENCY	45% 95%	optional	Station Group BDLC SWITCHED BDLC DEDICATED
FRSPTIMERLONG		required	BDLC Station Group
FRSTIMERSHORT		required	BDLC Station Group
HARDWARE		required	Station Level Manager
REMOTEBUSYMONITOR	reset	optional	BDLC Station Group
CHECKPOINTTIMER		required	BDLC Station Group
MAXSEGMENTSIZ	NWSS	optional	Station Level Manager
MONITOR	off	optional	Station Level Manager
REMOTEBUSYMONITOR	reset	optional	BDLC Station Group
RETRYDELAYTIMER		required	ACU - attribute
RETRYLIMITCHECKPT	3	optional	BDLC Station Group
RETRYLIMITSABMDIS	3	optional	BDLC Station Group
SELECTSTANDBY	off	optional	BDLC Station Group
SPEED		required	Station Level Manager
WINDOWSIZE	7	optional	BDLC Station Group

Figure 5-2. BDLC Station Level Attributes

STATION COMMANDS

Order of Entering Station Commands

As certain commands can refer to information of another command, related commands must be entered in the following order:

```
ADD PROFILE.  
ADD STATION.  
ADD ENSEMBLE.  
ADD CONNECTION.
```

For example, ensembles and connections need not be added unless they are used by the CALL commands, but if they are used, they must be added in the order shown. Profiles are only needed if they are required by subsequent ADD STATION commands, and then they must be added in the order shown.

With the exception of DELETE PROFILE, when deleting, these commands must be entered in reverse order. DELETE PROFILE may be deleted any time after a station has been added and there are no dependent add stations required.

Station Profile Commands

The Profile commands, ADD and DELETE PROFILE, add or delete a list of attributes known by a profile name.

Profiles are a collection of station attribute names and values which can be defined independently of any station, but which can be used later as a source of attribute values when stations are added to the node's network configuration.

Deleting a profile (and subsequently redefining it) has absolutely no effect on a station which was added using that profile. The profile is not a station attribute; it is merely a source of attribute values at the time a station is added.

No error checking about the validity of station attributes is made when profiles are added, other than the correctness of the attribute name and associated type of value. Appropriateness and validity checks are performed when the station is added and later used.

Command Syntax

```

ADD PROFILE <prof-name> (--- <sta-attrib.> = <value> ---) --|
-----
    
```

Response Syntax

```

<profile-name> ADDED AS PROFILE -----|
    
```

Semantics

This command adds the station attributes to the station level as a list of attributes known by the <profile-name>. This list can be used later as a source of attribute values when stations are added to the node's network configuration.

No error checking about the validity of station attributes is made when profiles are added, other than the correctness of the attribute name and associated type of value. Appropriateness and validity checks are performed when the station is added and later used.

Examples

```

ADD PROFILE BDLCTYPE (SPEED = 5000, EFF = 60, . . .)
=> PROFILE BDLCTYPE ADDED
    
```

Error Messages

- PROFILE ALREADY DEFINED
 - if a profile of the same name was previously defined.
- INVALID STATION ATTRIBUTE NAME
 - if an attribute is specified which is not a potentially settable station attribute.
- INVALID STATION ATTRIBUTE VALUE
 - if an attribute value of the wrong type is specified.

Command Syntax

```
DELETE PROFILE <profile-name> -----|  
---  ----
```

Response Syntax

```
<profile-name> DELETED AS PROFILE -----|
```

Semantics

This command removes the list of station attributes known as <profile-name>.

Deleting a profile has no effect on the station, or stations, added using that profile. The profile is not a station attribute; it is merely a source of attribute values at the time a station is added.

Examples

```
DELETE PROFILE BDLCTYPE  
=> PROFILE BDLCTYPE DELETED
```

Error Messages

```
PROFILE NOT DEFINED  
- if a profile is to be deleted which does not exist.
```

Add Station Command

Command Syntax

```

ADD STATION -- <sta-name> -- ( -- TYPE = --- BDLCDDEDICATED ----->
-----
                                     |-----|
                                     | - BDLCSWITCHEDACU - |
                                     |-----|
                                     | - BDLCSWITCHED -----|
                                     |-----|

>----->
|
| |<-----|
| |-----/2\-, -- PROFILE = <prof-name> -----|
| |-----|
|
|-----)-----|
|
| |<-----|
| |-----, - <attribute-name> = <value> -----|
| |-----|
|

```

Response Syntax

<station-name> ADDED AS STATION

Semantics

This command adds a BDLC station to the node's network configuration. The attribute values specified are for the station itself, plus those in the Station Level Manager station table. Zero, one or two profiles can be specified, and if specified, must appear before any attributes.

The values for the station attributes are taken from default values, for those attributes which have defaults. These values are then supplemented and possibly overridden by the values taken from one or two profiles, if specified, and finally by the attribute values supplied with the ADD STATION command. (Override applies if the same attribute is supplied from two or more sources.) If two profiles are specified, the one listed second supplements and possibly overrides the one listed first.

The ADD STATION command is used to add a BDLC station. The TYPE specified is either BDLC DEDICATED, BDLC SWITCHED with ACU, or BDLC SWITCHED.

BDLC DEDICATED is used for permanent connections.

BDLC SWITCHED with ACU is used for switched connections where the call is established by automatic dialout.

BDLC SWITCHED is used for switched connections in one of two ways: 1) to receive incoming calls, or 2) to establish an outgoing call by manual dialout.

The Operational commands, ESTABLISH CALL and AWAIT CALL describe the behavior of these kinds of stations in making calls with neighbor nodes.

Examples

```
ADD STA CAD (TYPE = BDLCDED, PROF = P1, PROF = P2)
```

```
=> CAD ADDED AS STATION
```

Error Messages

```
IMPROPER SETTING OF STATION ATTRIBUTE  
- if this attribute cannot be set.
```

```
NO MORE THAN TWO PROFILES MAY BE SPECIFIED
```

```
INVALID PROFILE REFERENCE  
- Profile not defined.
```

```
ATTRIBUTE INCONSISTENT WITH STATION TYPE  
- if attributes specified are inconsistent with the STATION TYPE (see  
Station Attributes below).
```

```
ALL REQUIRED ATTRIBUTES NOT SET
```

Delete Station Command

Command Syntax

```
DELETE STATION --- <station-name> -----|  
--- ---
```

Response Syntax

```
<station-name> DELETED AS BDLC STATION
```

Semantics

This deletes the specified station from the network configuration of the node. An error is returned and the delete does not take place if the station is part of an ensemble or connection, or if the station is in use for a call.

Examples

```
DELETE STA CAD  
=> CAD DELETED AS STATION
```

Error Messages

```
INVALID STATION  
- Station does not exist.
```

```
NOT DELETED, STATION IN USE
```

```
NOT DELETED, STATION IN ENSEMBLE
```

```
NOT DELETED, STATION IN CONNECTION
```

Modify Station Command

Command Syntax

```
MODIFY STATION  -- <sta-name> (--- <sta-att> = <value> ---) -|  
---            |<-----, -----|
```

Response Syntax

```
<station-name> MODIFIED AS STATION -----|
```

Semantics

This command permits certain attributes of an existing station to be changed. Only those attributes specified with this command are changed.

NOTE

Some attribute changes require that the station be disconnected, while others do not, and still others may never be changed.

Examples

```
MOD STA CAD (MONITOR = ON)  
=> CAD MODIFIED AS STATION
```

Error Messages

```
STATION ATTRIBUTE MAY NOT BE SET
```

STATION NOT CLOSED

- An attribute that can only be changed when station closed has been requested to be changed when station is open.

ENSEMBLE COMMANDS

Add Ensemble Command

Command Syntax

```
ADD ENSEMBLE --- <ensemble-name> ----->
-----
```

```

      |<----- , ----->|
>--- INCOMING --- ( --- <bdlc-switched-sta-name> --- ) ---|
|   --          |
| - OUTGOING - |
|   ---          |
      |<----- , ----->|
| - PERMANENT -- ( --- <bdlc-ded-sta-name> --- ) --- |
|   ----          |

```

Response Syntax

<ensemble-name> ADDED AS ENSEMBLE

Semantics

This command defines an ensemble of stations. Ensembles so defined can later be referenced in defining connections and in establishing calls.

An ensemble can be either incoming, outgoing, or permanent. The stations in the ensemble must be one of the following:

All BDLC switched (for in or out ensembles).
All BDLC permanent.

Examples

```
ADD ENS NITELINES IN (STAONE, STATWO)
ADD ENS MAINLINE PERM (XPVA, XPVB, XPVC)
ADD ENS GROUPLINE OUT (GROUPX)
```

Error Messages

ENSEMBLE ALREADY ADDED

STATIONS IN ENSEMBLE HAVE NOT BEEN ADDED

STATIONS IN ENSEMBLE NOT OF PROPER TYPE

Delete and Modify Ensemble Commands

Command Syntax

```
DELETE ENSEMBLE <ensemble-name> -----|  
---   ---
```

Response Syntax

```
<ensemble-name> DELETED AS ENSEMBLE
```

Semantics

The DELETE ENSEMBLE command removes the ensemble definition from the Station Level.

The DELETE command may not be entered when the ensemble is being used by an existing call, or if the ensemble is being referenced by an existing connection.

Examples

```
DEL ENS NITELINES  
=> NITELINES DELETED AS ENSEMBLE
```

Error Messages

```
ENSEMBLE IN USE, MAY NOT BE DELETED
```

```
NO SUCH ENSEMBLE
```

Command Syntax

```
MODIFY ENSEMBLE <ens-name> --- + -- ( --- <sta-name>--- )--|  
--- --- |<--- , --->|  
|---|  
|---|
```

Response Syntax

<ensemble-name> MODIFIED AS ENSEMBLE

Semantics

The MODIFY ENSEMBLE command adds or deletes stations from the ensemble. The last remaining station in an ensemble cannot be deleted by use of MODIFY ENSEMBLE; DELETE ENSEMBLE must be used for this.

Examples

```
MODIFY ENS NITELINES + (STASIX, STAFOUR)  
=> NITELINES MODIFIED AS ENSEMBLE
```

Error Messages

DELETED STATION NOT IN ENSEMBLE

LAST REMAINING STATION MAY NOT BE DELETED

ADDED STATION(S) OF IMPROPER TYPE

NO SUCH ENSEMBLE

NEIGHBOR CONNECTION COMMANDS

These commands provide station related information for later Establish Call commands, or for implicit (automatic) call establishment.

In the following command syntaxes, <neighbor> is:

```

----- <host-name> -----|
|                               |
| - #<node address> -----|
|                               |
| - <host-name> #<node-address> -|

```

Add Connection Command

Command Syntax

```

ADD CONNECTION --- INCOMING --- WITH --- <neighbor> ----->
-----
|                               |
| - OUTGOING --|
|                               |
| - PERMANENT -|
|                               |
|-----|

```

```

>--- BY ----- <station-name> ----->
---
|                               |
| - STATION -|
|                               |
| - ENSEMBLE --- <ensemble-name> -----|
|                               |
|-----|

```

```

>-----|
|                               |
| <-----, -----|
| - (----- INITQUANTITY = <integer> -----) -|
|                               |
| - CALldata = ---- <bdlc-call-data> -|
|                               |
|-----|

```

Response Syntax

CONNECTION ADDED WITH <neighbor>

Semantics

The ADD CONNECTION command is used to define potential connections between the stations of the local node and its neighbors. The command does not directly open any actual connections. It is used to provide information for automatic call establishment, and for subsequent ESTABLISH CALL, and AWAIT CALL commands.

There are three directions, or types of connections: permanent, incoming, and outgoing. Permanent connections apply only to dedicated type stations; incoming and outgoing connections apply to BDLC switched stations. Both incoming and outgoing may be defined for a switched station. One or more connections may be defined for a given neighbor.

The ADD CONNECTION commands for permanent and incoming connections are similar in effect. If a station has connections defined with certain nodes, then only those nodes will be permitted to establish connections via that station. If the station has no connections defined, then no such restrictions will be applied (unless the <neighbor> information is supplied with an ESTABLISH CALL command). Incoming connections apply only to incoming calls received on a switched station. A call is considered to be incoming if the AWAIT CALL command was issued for the station, or if a call was initiated due to a station having AUTOINIT true. A call is considered outgoing if the ESTABLISH CALL was used. There is a case where a call is started as outgoing, but becomes incoming due to a call collision and the acceptance of the incoming call.

The ADD CONNECTION commands for outgoing connections are somewhat different. For switched stations with an ACU, they primarily provide a means of storing a telephone number used to call a particular node. They can also be used to request that, at initialization time, a call automatically be placed to that node, by the specified station. These two features are available through the use of the attributes INITQUANTITY and CALldata.

INITQUANTITY represents the number of calls that should be placed, at initialization time, by the specified station, or ensemble of stations. For a single station, the only valid values are zero (0) or one (1). For an ensemble, the valid values are zero to the number of stations in the ensemble. The default value in either case is zero.

CALldata is a telephone number. Where multiple calls are made to a neighbor, in the case of an ensemble of stations, the neighbor telephone number must represent the "hunting" telephone number. A hunting telephone number is a feature provided by many telephone services. It allows a group of lines to be called using a single telephone number, and only when all numbers are in use will the call be found busy. For further detail on the telephone number, refer to BDLC CALL DATA, in Appendix A.

*Semantics Summary***INCOMING connections:**

Incoming connections may be specified by using a BDLC switched station, or an ensemble of BDLC switched stations.

OUTGOING connections:

Outgoing connections may be specified using a BDLC switched station with ACU or an ensemble of such stations.

INITQUANTITY may be specified; this determines the number of outgoing calls attempted automatically at initialization. The default value is zero.

CALL DATA must be specified for outgoing calls on BDLC stations with ACU.

PERMANENT connections:

Permanent connections specify either a dedicated BDLC station, or an ensemble of such stations.

Examples

```
ADD CONN IN WITH CENTRAL #5 BY STATWO
```

```
ADD CONN OUT WITH CENTRAL BY ENS NITELINES  
(CALLDATA = "1-215-492-1234", INITQUANT = 1)
```

*Error Messages***IMPROPER ENSEMBLE KIND**

- if ensemble is not of correct kind (IN, OUT, or PERM).

NEIGHBOR IS NOT A NODE

- if ROUTER VAL is true and neighbor was not added as a node.

IMPROPER STATION KIND

- if station type is not compatible with connection type.

INCORRECT PARAMETERS

- if INIT QUANT or CALL DATA is specified for PERM or IN connections, or CALL DATA omitted for OUTGOING connections.

Delete and Modify Connection Commands

Command Syntax

```
DELETE CONNECTION --- <connection-definition> ----->
---
```

```
>-- BY ----- <station-name> -----|
--  |         |         |         |         |
--  | - STATION - |         |         |         |
--  |   ---     |         |         |         |
--  | - ENSEMBLE --- <ensemble-name> -----|
--  |         |         |         |         |
--  |         |         |         |         |
```

Response Syntax

```
CONNECTION DELETED WITH <neighbor>
```

Semantics

The DELETE command deletes a specified connection with a neighbor, provided that connection is not in use by a call.

If the WITH <neighbor> phrase is omitted, connections with all neighbors that are OUTGOING are deleted, provided none of these are in use.

Examples

```
DEL CONN IN WITH CENTRAL #5 BY STATWO
=> CONNECTION DELETED WITH CENTRAL #5
```

Error Messages

```
NO SUCH CONNECTION
- if the connection specified (by the connection definition) was not added.
```

```
CONNECTION IN USE
- if the connection is in use for a call.
```

Command Syntax

```

MODIFY CONNECTION --- OUTGOING ----->
---      ---      | - WITH -- <neighbor> -- |
                        -----

>--- BY -----<station-name> ----->
---      | - STATION - |
          | - ENSEMBLE --- <ensemble-name> ----- |
          -----

>----- (----- INITQUANTITY = <integer> -----) ----- |
          | - CALldata = <bdlc-call-data> - |
          -----
    
```

Response Syntax

CONNECTION MODIFIED WITH <neighbor>

Semantics

The MODIFY command modifies a specified connection with a neighbor.

The connection information, INITIAL QUANTITY and CALL DATA, can be changed by the MODIFY CONNECTION command. However, if the connection is in use, the changes will only take effect after the connection has been closed.

Examples

```

MOD CONN OUT WITH CENTRAL BY ENS NITELINES (INITQUANT = 2)
=> CONNECTION MODIFIED WITH CENTRAL #5
    
```

Error Messages

NO SUCH CONNECTION

CONNECTION IN USE

IMPROPER PARAMETER(S) SPECIFIED

- if the INIT QUANT or CALL DATA is incorrectly specified for the connection (note that no parameters exist for INCOMING or PERMANENT connections, so these connections cannot be modified).

SECTION 6

OPERATIONAL COMMANDS

Operational Commands are commands which cause node actions that relate to the network, by making or breaking neighbor connection, by breaking a host-to-host (port-level) communication, or by performing special routing actions.

The Operational commands can be entered only during the operating phase of Network mode.

NOTE

To establish and break communications with a Program Agent are operational functions. However, the operational command to perform these functions, PROGRAM, is not discussed in this section. For detail of the PROGRAM command, refer to Section 3.

CALLS WITH NEIGHBOR COMMANDS

These commands affect the communications to a neighbor. The commands are: ESTABLISH CALL, AWAIT CALL, CLEAR CALL, SAVE STATION, and READY STATION.

AWAIT CALL COMMAND

Command Syntax

```
AWAITCALL BY ----- <station-name> -----|
- - - - - |
| - STATION - |
| --- |
| - ENSEMBLE --- <ensemble-name> ----- |
| --- |
```

Response Syntax

CALLS WAITING BY STATION <station-name>

CALLS WAITING BY ENSEMBLE <ensemble>

Semantics

This command places the indicated station or all available stations in the indicated ensemble or station group into a state to receive a call.

AWAIT CALL cannot be used for permanent stations or ensembles of permanent stations.

The station, if specified, must be a switched BDLC station. The ensemble, if specified, must be an incoming ensemble of BDLC switched stations.

The neighbor who initiates the call is validated if a corresponding connection has been added.

An implicit AWAIT CALL occurs during initialization or when a station is made READY, if AUTO INIT is true.

Examples

AWAITCALL BY HOTLINES

AC BY ENS GROUPX

Error Messages

NO SUCH ENSEMBLE

- if ensemble does not exist.

IMPROPER ENSEMBLE TYPE

- if ensemble is of incorrect type.

NO SUCH STATION

NO SUCH STATION GROUP

IMPROPER STATION TYPE

- if BDLC station is of wrong type.

NO STATION AVAILABLE

- if all stations in the specified ensemble are unavailable.

STATION NOT AVAILABLE

- if the station specified is unavailable (that is, busy or saved).

VALIDATION FAILURE

- if neighbor called has failed validation.

STATION FAILURE: <reason>
- if BDLC station reports a failure.

CLEAR CALL COMMAND

Command Syntax

```
CLEARCALL --- WITH --- <neighbor> -----|
-      -      |-----|
-      -      | - BY ----- <station-name> -----|
-      -      | ---|
-      -      | - STATION -|
-      -      | ---|
-      -      | - ENSEMBLE ---- <ensemble-name> -----|
-      -      | ---|
```

Response Syntax

CALLS CLEARED WITH <neighbor>
CALLS CLEARED BY . . . (as specified in command above)

Semantics

This command clears one or more calls to a neighbor.
If the neighbor is specified, all calls to that neighbor are cleared.
If a station is specified, the call on that station is cleared.
If an ensemble is specified, all stations in that ensemble are cleared.

Examples

```
CLEARCALL WITH BRANCH2
CC BY ENS NITELINES
```

Error Messages

STATION ALREADY CLOSED

NO CALLS WITH NEIGHBOR EXIST

ALL STATIONS IN ENSEMBLE ARE CLOSED

NO SUCH STATION

NO SUCH ENSEMBLE

ESTABLISH CALL COMMAND

Command Syntax

```

ESTABLISHCALL ----->
-      -      |--- WITH <neighbor> ---|
                -----

>-- BY ----- <station-name> ----->
--  |         |
--  | - STATION - |
--  |   ---     |
--  | - ENSEMBLE --- <ensemble-name> ----- |
--  |         |
--  |         |

>-----|
      |--- (CALLDATA = <bdlc-call-data>) ---|
      -----
    
```

Semantics

This command establishes one or more new calls to neighbor nodes. The forms of this command, and the functions performed are described as follows:

ESTABLISHCALL BY permanent station:

ESTABLISHCALL WITH neighbor BY permanent station:

Establish a new call with this neighbor using the specified station. The WITH <neighbor> clause is optional; if supplied, the identity of the neighbor is validated.

ESTABLISHCALL BY permanent ensemble:

Create one or more new calls using all of the stations in the permanent ensemble. Validate the neighbor identities if an appropriate connection was previously entered. The WITH <neighbor> clause must not appear.

ESTABLISHCALL WITH neighbor BY auto-dial station:

ESTABLISHCALL WITH neighbor BY auto-dial station
(CALLDATA = <bdlc-call-data>):

Establish a new call with this neighbor using the specified station. The station must be of type BDLC-switched-with-ACU. The WITH <neighbor> clause must appear, and the neighbor identity is validated. CALLDATA is optional; it is discussed below.

ESTABLISHCALL WITH neighbor BY outgoing ensemble:

ESTABLISHCALL WITH neighbor BY outgoing ensemble
(CALLDATA = <bdlc-call-data>):

Establish a new call with the neighbor using the first available station in the outgoing ensemble. (The ensemble contains BDLC switched stations with ACU.) The WITH <neighbor> clause must appear, and the neighbor identity is validated. CALLDATA is optional; it is discussed below.

ESTABLISHCALL WITH neighbor BY manual-dialout station:

Establish a new call by setting this station, which is a BDLC switched station without ACU, so that it can be used for manual dialout. The operator uses this station to manually dial a neighbor. The identity of the neighbor is validated. CALLDATA must not appear.

When CALL DATA may be used:

Call data applies only to a switched BDLC station with ACU, or to an outgoing ensemble of such stations. Where it applies, the rule for using call data is as follows:

If, previously, an outgoing connection to that neighbor was added, CALldata information must not appear with this command (since it was supplied by the connection entry). If no such connection was previously added, the CALldata information must be supplied with this command, and applies only for the duration of this call.

Automatic Establish Calls at Initialization:

During initialization, an ESTABLISH CALL is generated for a permanent station for which AUTO INIT is set to TRUE. Further, for switched outgoing connections in which INIT QUANTITY is set to a nonzero value, n, then during initialization, n number of ESTABLISH CALLs are generated using the stations specified in that connection.

Response Syntax

CALL ESTABLISHED BY <station or ensemble>

This response is returned when a call establishment by a single station is requested and succeeds immediately. A call establishment by a single station is requested by specifying one of the following in the ESTABLISH CALL command:

- . A permanent station, BDLC DEDICATED.
- . A BDLC switched station, with ACU for auto-dial, or without ACU for manual dial.
- . An outgoing ensemble, used to select a switched station.

CALL INITIATED BY <permanent station>

This is returned when a permanent station is specified and no error is encountered, but the corresponding remote station is in a pending open state. Note that an ATTACH report is given later when this remote station establishes the other end of the call.

CALLS IN PROCESS BY <permanent ensemble>

This is returned when an ensemble of permanent stations is specified. In this case, an immediate response is given indicating that the command has begun to be serviced. Then later, as a call is being established with EACH

station in the ensemble, a report is given. This report indicates if the call establishment is completed (an ATTACH report), if the call is initiated (as previously described for single permanent stations), or if an error has occurred (a station failure report).

Examples

ESTABLISHCALL WITH BRANCH2 BY STATWO

EC WITH BRANCH2 BY ENS NITELINES (CALldata = "12145551212")

Error Messages

NO SUCH ENSEMBLE

- if ensemble does not exist.

IMPROPER ENSEMBLE TYPE

- if ensemble is of incorrect type.

NO SUCH STATION

IMPROPER STATION TYPE

- if BDLC station is of wrong type.

NO STATION AVAILABLE

- if all stations in the specified ensemble are unavailable (unavailable means the station is busy with a call or is SAVED).

STATION NOT AVAILABLE

- if the station specified is unavailable.

VALIDATION FAILURE

- if neighbor called has failed validation.

STATION FAILURE: <reason>

- if BDLC station reports a failure.

CALL DATA REQUIRED

CALL DATA NOT REQUIRED

SAVE AND READY STATION COMMANDS

Command Syntax

```
READY ----- STATION ----- <bdlc-station-name> -----|  
-----
```

Response Syntax

```
STATION <station-or-group-name> READIED
```

Semantics

The READY STATION command does the following: if the Station is connected, it removes the to-be-saved mark; if the Station is saved, it makes the station available again. In the latter case the call is re-established if AUTO INIT is true.

Error Messages

INVALID STATION NAME

- if Station does not exist or syntactically invalid.

Command Syntax

```
SAVE ----- STATION ----- <bdlc-station-name> -----|  
-----
```

Response Syntax

```
STATION <station-or-group-name> TO BE SAVED
```

Semantics

The SAVE STATION command does the following: if the station is connected, it marks the station to be saved later, when it becomes disconnected (by a call terminating or the station being cleared). If the station is already disconnected, this command takes immediate effect and the station becomes unavailable to be connected (that is, SAVED).

Error Messages

```
INVALID STATION NAME
```

- if Station does not exist or syntactically invalid.

HOST-TO-HOST COMMUNICATIONS

These commands concern host-to-host communications. They are CLEAR HOST, SAVE HOST, and READY HOST.

Command Syntax

```
CLEAR ---- HOST <host-name> -----|  
-----|  
| - #<node-address> - |
```

Response Syntax

```
HOST <host-name> #<node-address> ----- CLEARED -----|
```

Semantics

The CLEAR HOST command abruptly severs all user communication to a host, but new communications can then begin. That is, open or opening subports are immediately deactivated. However, Port Level communication is not severed; and after the CLEAR, unless a previous SAVE was done, new user subports can be opened.

Error Messages

NO SUCH REMOTE HOST
- if remote host is unknown to the Port Level.

Command Syntax

```
READY ---- HOST <host-name> -----|  
-----|  
| - #<node-address> - |
```

Response Syntax

```
HOST <host-name> #<node-address> ----- READIED -----|
```

Semantics

The READY HOST command removes the SAVE status of the remote host, and the remote host, when present, is available for opening new subports.

Error Messages

NO SUCH REMOTE HOST
- if remote host is unknown to the Port Level.

Command Syntax

```
SAVE ----- HOST <host-name> -----|  
----      ----      |                |  
|- #<node-address> -|
```

Response Syntax

```
HOST <host-name> #<node-address> - SAVED, SUBPORTS OPEN: <intgr> --|
```

Semantics

The SAVE HOST command marks the remote host TO BE SAVED. No new subports are opened and any pending opens are aborted. As soon as the last subport to that host is closed, it is SAVED. No further subports can then be opened to that host while in the SAVED state. (If there are no subports in use to that host when the SAVE is entered, the host is immediately SAVED.)

Error Messages

NO SUCH REMOTE HOST
- if remote host is unknown to the Port Level.

OTHER OPERATIONAL COMMANDS

The commands that follow are commands that do not fall under any particular category. Thus they are just grouped here for convenience.

LOG COMMENT COMMAND

Command and Response Syntax

LOGCOMMENT <text>

- - -

=> COMMENT LOGGED

Semantics

This command causes text to be logged as a Network Services comment. It will be logged along with other Network Services logging information.

ROUTING REFRESH COMMAND

This command can be used to cause the local Router to update its routing tables. It does this by requesting routing information from its neighbors.

Command Syntax

ROUTINGREFRESH

- - - - -

Response Syntax

ROUTING INFORMATION REFRESHED

Semantics

This command causes the Router to request all its neighbors to send NETCHANGE messages, from which the local routing information is updated.

START TRACE COMMAND

Command Syntax

```
STARTTRACE <trace-source> TO <trace-destination>
```

where <trace-source> and <trace-destination> are:

```
----- <host-name> -----|
| - #<node address> -----|
| - <host-name> #<node-address> -|
```

Response Syntax

```
TRACE STARTED: <trace-source> TO <trace-destination>
```

Semantics

This command causes the Router to issue a ROUTER CONTROL FRAME to either do a trace start (if the local node is not the trace source) or a trace (if it is). The response to this command is simply an acknowledgement that the Router has sent the frame.

When trace results are received, Routing Trace Reports are received from all involved nodes.

Examples

```
START TRACE BRANCH1 TO BRANCH2
```

```
=> TRACE STARTED: BRANCH1 #3 TO BRANCH2 #5
```

Error Messages

```
INVALID TRACE SOURCE
```

```
- if ROUTER VALIDATE is true and source node is invalid.
```

```
INVALID TRACE DESTINATION
```

```
- if ROUTER VALIDATE is true and destination node is invalid.
```

```
UNREACHABLE TRACE SOURCE
```

```
UNREACHABLE TRACE DESTINATION
```

SENDING TEXT TO AN AGENT

This command is intended to be used for sending text messages to another Network Services operator (agent). In addition, it can be used as a maintenance tool for basic testing of the network.

Command Syntax

```
TO ----- # <node-address> --- : <text> -----|
-- |
  | - <host-name> - |
  | ----- <host-name> ----- |
```

Response Syntax

```
TEXT SENT TO <host-name> #<node-address>
```

Semantics

This command sends text to another node. The arrival of the text causes a FROM report (message from remote agent report). The text must be containable within a single Router Frame; hence, the text and associated frame header cannot exceed MAX SEGMENT SIZE in use to the receiving node.

Note that the text may contain any sequence of characters except ";" (the command terminator) or ":" (the response suffix separator), unless these are enclosed in quoted strings.

Error Messages

```
NO SUCH HOST OR NODE
```

```
NODE UNREACHABLE
```

```
MESSAGE TEXT EXCEEDS MAX SEGMENT SIZE
```

OPEN CPD/SD
CLOSE CPD/SD
VALIDATEANDATTACH
DETACH
SENDTEST

SECTION 7 STATION CONFIDENCE (MANUAL) COMMANDS

Station Confidence commands are used to establish confidence in the proper functioning of the stations. Together with inquiry commands, they can be used to diagnose problems. They can be entered at any time after the Configuration Entry phase of network initialization, and used on stations that are not SAVED.

Command Syntax

```
OPEN ----- CPD --- <station> -----|
-----|
CLOSE --| | - CALLDATA = <bdlc-call-data> -|
-----|
|
|-- SD --- <station> -----|
|--
```

% calldata option for
OPEN only

(CPD = CONNECTION PORT DIALOG)
(SD = STATION DIALOG)

```
VALIDATEANDATTACH ----- <station> -----|
---|
DETACH -----|
---
```

```
SENDTEST --- <station> -----|
-----|
|
|----- <string> -----|
| -- HEX <hex-number> -|
|-----
```

**OPEN CPD/SD
CLOSE CPD/SD
VALIDATEANDATTACH
DETACH
SENDTEST**

*Section 7
Station Confidence
(Manual) Commands*

Semantics

These commands cause manual command mode messages to be passed through the Station Level manager to the Stations themselves.

OPENing CPD (Connection Port Dialogue) opens the station to the extent that test messages can be sent. No BDLC protocol is available. Specifying CALldata on a switched BDLC station with ACU causes the station to dial the telephone number provided.

OPENing a station for SD (station dialogue) opens the station and initializes the BDLC protocol. Test messages can be sent. If the station is defined as outgoing, the station commences sending SABM to establish the BDLC connection.

VALIDATE AND ATTACHing a station causes the station to send out greetings. If the greetings complete successfully, the station is connected to the Router for normal traffic.

DETACHing a station severs the connection between the station and the router, and leaves the station in a state that can be used for testing. If the station was in use, the station level report to the router causes the station link to be removed from any routing.

CLOSEing a SD (Station Dialogue) terminates the station protocol (for example, BDLC) function. The station will not react in any way to input data. It will, however, react to interface changes as recognized by the Connection Port (for example, it will answer an incoming call, but will not commence any dialogue).

CLOSEing CPD (Connection Port Dialogue) causes the station to become dormant. It will not react in any way to changes on the link interface.

SENDTEST can be used to send a test message to another node. The message is sent by the SLM (Station Level Manager) to another node SLM. At the receiving SLM, the message is reported to the Operations Interface and it is retransmitted back to the originating node. The originating SLM receives the echoed message and sends a report to the Operations Interface.

The reports displayed are Test Received (remote node) and Test Echoed (local node). Refer to Section 10 for details of these station reports.

The test messages can be sent when the station is being used for normal traffic, and when in manual mode, where the station must be open for Connection Port Dialogue.

NOTE

To manually attach a station to the Router the station must be opened for station dialogue, and for opening station dialogue it must be open for Connection Port Dialogue.

Response Syntax

MANUAL COMMAND SERVICED ON <station>

Also refer to Station Reports for the reports that are generated due to the above actions.

Error Messages

BDLC FAILURE ON <station> -

VALIDATION FAILURE ON <station> -
this occurs in response to the VALIDATE AND ATTACH command.

INVALID SLM PHASE

INVALID STATION PHASE ON <station>

ERROR, AUTO COMMAND IN PROGRESS

COMMAND OUT OF SEQUENCE

SECTION 8

SERVICE COMMANDS FOR OIM INPUT FILES

Service commands are used to control the activity of OIM (Operations Interface Messages) files and to specify the disposition of responses to Operations Interface commands. They can be entered at any time.

LOAD COMMAND

Command Syntax

```
LOAD ----- <file-title> -----|
-----
                        |-----|
                        |--- SYNTAX ---|
                        |-----|
                        -----
```

Semantics

The LOAD version of this command reads the file and services the commands in the order presented; if a syntax error is encountered, the reading of the file is aborted.

The LOAD FOR SYNTAX version scans the file for syntax errors only; the commands are not otherwise serviced.

Normally, the input file is not an initialization file because such a file is read automatically when Network Services is initiated. However, it is possible to LOAD a file while the Node is still initializing. In such an event, the input file is treated as a secondary initialization file. If this technique is used to enter additional parameters for initialization, the ENDINIT command can be included in the secondary initialization file.

LOAD

This command can be entered only from an ODT or Program Agent. Only one OIM input file can be in the process of being loaded at any one time.

Examples

LOAD CHANGEFILE

=>LOADING CHANGEFILE

Response Syntax

```
LOADING ----- <file-title> -----|
      | - FOR SYNTAX - |
```

Error Messages

<file-title> NOT PRESENT FOR LOADING

IMPROPER FILE FORMAT

- if file is not formatted as specified for an OIM input file.

FILE LOADING IN PROCESS

- if an OIM input file is already being loaded or is suspended.

STOP COMMAND

Command Syntax

```
STOP -----|  
----
```

Response Syntax

```
-- <file-title> STOPPED -----|  
                                | - AT <sequence-number> - |
```

If the sequence number appears, it indicates the last command serviced.

Semantics

This command causes the OIM input file, currently being loaded, to be suspended. The sequence-number displayed indicates the sequence number (if available) of the next command to be executed. This file can be resumed at the point where it was suspended (by a CONTINUE command) or the loading can be aborted (by a DISCARD command). While the file is suspended, other commands may be entered by the ODT or Program Agents.

If the file being read contains a STOP command, it suspends itself.

Program Agents cannot enter this command.

Examples

```
STOP  
=> CHANGEFILE STOPPED  
=> CHANGEFILE STOPPED AT 2351000
```

Error Messages

NO FILE BEING LOADED

FILE ALREADY SUSPENDED

CONTINUE

CONTINUE COMMAND

Command Syntax

```
CONTINUE -----|  
-----
```

Response Syntax

```
<file-title> CONTINUED
```

Semantics

This command causes a suspended OIM input file to resume loading. If the file was suspended because of an error, the file input resumes immediately after the erroneous command which caused the suspension. If the file was suspended by a STOP command, the file is resumed at the point of suspension.

This command, entered from an input file, has no effect. It cannot be entered from a Program Agent.

Example

```
CONTINUE
```

```
=> CHANGEFILE CONTINUED
```

Error Messages

```
NO SUSPENDED FILE
```

DISCARD COMMAND

Command Syntax

```
DISCARD -----|  
-----
```

Response Syntax

```
<file-title> DISCARDED
```

Semantics

This command aborts the loading of an OIM input file which has been suspended. This command does not discard a file if it is actively being loaded. (To do the latter, a STOP command followed by the DISCARD command should be entered.)

Example

```
DISCARD  
=> CHANGEFILE DISCARDED
```

Error Messages

```
NO SUSPENDED FILE
```

SECTION 9 SUMMARY AND DETAIL INQUIRIES

This section describes inquiries that provide information about the Network Services operation. Inquiries can be detailed or summarized. Each level has its own view of the network, and the inquiries provide access to this information. This section is organized into descriptions of inquiry types: Host, Node, Neighbor, Routing, Profile, Station, Ensemble, Candidate, Port Counts, and Modes.

In this section the inquiry command format is shown followed by the response syntax only. In most cases there are several forms of response, and each is described individually. There are no error messages except for the syntax errors described in Section 1.

HOST SUMMARY AND DETAIL

Command Syntax

```

HOST -----|
-----|
      | -- <host-name> -- |
    
```

The HOST form of this inquiry returns a list of hosts with summary information about each.

The HOST <host-name> form returns detailed information about the host selected.

Response Syntax

The detailed responses are described in the following subsections, Host Summary and Host Detail.

HOST SUMMARIES

Using the "HOST" form of the HOST command results in the response:

```

      |-----,-----|
=> HOST --- <host> (--- UNAVAILABLE -----) ---|
      |
      |         |-----,INVALID-----|
      |         |-----|
      |         | INACTIVE -----|
      |         | GREETING -----|
      |         | TERMINATING -----|
      |         | INTERRUPTED -----|
      |         | ACTIVE -----|

```

where <host> is

```

--- <host-name> #<node-address> -----|

```

This summary displays the state of each host that is in communication with the local host. It reflects the states of the port levels in each host.

INVALID is returned when the host is marked "FOR TRANSLATE ONLY".

HOST DETAIL

Using the "HOST <host-name>" form of the Host command results in the response:

```
=> HOST ----| <host-name> #<node-address> ----->

>- ( --- UNAVAILABLE ----- ,READY,---->
    |                                     | - ,INVALID - | - ,SAVED,- |
    | INACTIVE ----- |
    | GREETING ----- |
    | TERMINATING ----- |
    | INTERRUPTED ----- |
    | ACTIVE ----- |

    |<-----,-----|
>----- MAX SEGMENT SIZE --- = --- <integer> ----- )-|
    |                                     | - *UNKNOWN -- |
    | SUBPORTS IN USE ---- = <integer> ----- |
    | PROTOCOL LEVEL IN USE = <integer> ----- |
```

This detailed summary provides enhanced host information about a specific remote host.

NODE SUMMARIES

```
NODE -----|  
----
```

This inquiry provides information regarding the reachability of all known nodes.

Response Syntax

```
      |<----- , -----|  
NODE ----- <neighbor> -----|  
      | - : --- REACHABLE --- |  
      | - UNREACHABLE - |
```

NEIGHBOR AND CONNECTION INFORMATION

```
NEIGHBOR -----|  
----|  
    | --- ACTIVE -----|  
    | ---|  
    | --- CONNECTION ----- <neighbor> ----|  
    | ---|  
    | --- DETAIL -----|  
    | -----|
```

"NEIGHBOR" or "NEIGHBOR ACTIVE" returns a list of neighbors and summary information about each (ACTIVE specifies only active neighbors; the unqualified form specifies all neighbors, potential and active).

"NEIGHBOR CONNECTION <neighbor>" returns connection information with the specified neighbor.

"NEIGHBOR DETAIL <neighbor>" returns detailed information about the specified neighbor.

Response Syntax

The following subsections Neighbor Summaries, Neighbor Connection Information, and Neighbor Detail describe the form of the responses.

Neighbor Summaries

Using the "NEIGHBOR" or "NEIGHBOR ACTIVE" entry of the NEIGHBOR command results in the response:

```

      |<-----,-----|
=> NEIGHBOR --- <host-n> #<node-a> --- DISCONNECTED ---|
      |              |              |
      |              | - BY <sta-list> -|

```

where <station-list> is:

```

      |<-----,-----|
----- <bdlc-sta-name> : --- BDLIC DEDICATED -----|
      |              |
      |              | - BDLIC SWITCHED -----|
      |              | - BDLIC SWITCHED ACU -----|

```

DISCONNECTED is returned when the NEIGHBOR STATUS is CLOSED. If ACTIVE is specified, these neighbors are not listed.

Neighbor Connection Information

Using the "NEIGHBOR CONNECTION <neighbor>" form of the NEIGHBOR command results in the response:

```

      |<----- , -----|
=> CONNECTION WITH <neighbor> --- <connection-info> ---|

```

where <connection-information> is:

```

----- INCOMING --- BY----- STATION <station-name> ----->
| - OUTGOING - | | - ENSEMBLE --- <ensemble-name>- |
| - PERMANENT- | |

```

```

>----->
| |<----- , -----| |
| -- (----- INITQUANTITY = <integer> -----) - |
| | - CALldata = ----- <bdlc-call-data> - |

```

```

>-----|
| |<----- , -----| |
| - ATTACHED BY ----- <station> -----|

```

where <station> is defined as:

```

----- <bdlc-station-name> : --- BDLC DEDICATED -----|
| - BDLC SWITCHED -----|
| - BDLC SWITCHED ACU -- |

```

Neighbor Detail

Using the "NEIGHBOR DETAIL <neighbor>" form of the NEIGHBOR command results in two types of response. Neighbor Connected, or Not Connected determines the type of response. A description of both forms of the response follows:

If neighbor is connected:

```

=> NEIGHBOR <host-name> #<node-address> ----->
      |<-----,-----|
>-- (----- WORKING RESISTANCE FACTOR = <integer> -----)>--
      |
      | - REMOTE RESISTANCE FACTOR = <integer> -
      | - LOCAL RESISTANCE FACTOR = <integer> -
      | - OPERATIONS RESISTANCE FACTOR = <integer> -
      | - MAX SEGMENT SIZE IN USE = <integer> -
      |
      |<-----,-----|
>-- LINKED ON --- <sta> (----- SPEED ----- = <integer> -----)>--
      |
      | - EFFICIENCY = <integer>--
      | - MAX SEG SIZE= <integer>--
    
```

where <station> is defined as:

```

--- <bdlc-station-name> : --- BDLIC DEDICATED -----|
      |
      | - BDLIC SWITCHED -----|
      | - BDLIC SWITCHED ACU -----|
    
```

If neighbor is not connected:

```

=> NEIGHBOR <host-name> #<node-address> NOT CONNECTED
    
```

ROUTING INFORMATION

This inquiry provides local routing table information.

Command Syntax

```
ROUTINGS -----|
-----|
| - INUSE - | | - TO ----- <host-name> -----|
| ----- | | --|
| - VIA - | | - #<node-address> -|
| ----- | | --|
| - *ALL -----|
| -----|
```

Response Syntax

Several types of response syntax are provided for this inquiry. The following subsections, Responses for Destination Inquiries (TO <node>), and Responses for Neighbor Routings (VIA <node>) describe these responses.

In each of the responses, <routing-info> is:

```
|<----- , -----|
-----|
| INUSE -----|
| |
| -- RESISTANCE FACTOR = <integer> -|
| -- HOP COUNT ----- = <integer> -|
| -- MAX SEGMENT SIZE = <integer> -|
```

Note that INUSE is present only if true for the given routing.

Also in the following responses, <node> is:

```
--- <host-name> #<node-address> ---|
```

Responses for Destination Inquiries (TO <node>)

Using the "ROUTINGS" or "ROUTINGS TO ALL" form of the ROUTING command results in the response:

```
=>
      |-----;-----|
      |-----,-----|
ROUTINGS---TO <node> --- VIA <node> --- UNREACHABLE ---|
      |-----|
      |----- (<rout-info>)-|
      |----- UNREACHABLE -----|
```

These commands return all routings for every destination node, arranged by destination node, and then, for each destination node, all of the neighbor nodes potentially usable to reach that destination.

Using the "ROUTINGS TO <node>" form of the ROUTINGS command results in the response:

```
=>
      |-----,-----|
ROUTINGS TO <node> --- VIA <node> --- UNREACHABLE ---|
      |-----|
      |----- (<rout-info>)-|
      |----- UNREACHABLE -----|
```

This displays all routings for a specified destination node, displaying each neighbor node which is potentially usable to reach that destination.

Using the "ROUTINGS INUSE" or "ROUTINGS INUSE TO ALL" form of the ROUTINGS command results in the response:

```

=> INUSE ROUTINGS --- TO <node> --- VIA <node> (<rout-info>) ---|
      |<----- ; -----|
      |----- UNREACHABLE -----|
  
```

These return in-use routings for every reachable destination node, showing the neighbor node currently used to reach each destination.

Using the "ROUTINGS INUSE TO <node>" form of the ROUTINGS command results in the response:

```

=> INUSE ROUTING TO <node> --- VIA <node> (<rout-info>) ---|
      |----- UNREACHABLE -----|
  
```

This returns in-use routing for a specified destination. If destination is unreachable (no in-use routing exists) then UNREACHABLE is displayed.

Responses for Neighbor Routings (VIA <node>)

Using the "ROUTINGS VIA ALL" form of the ROUTINGS command results in the response:

```

=>
      |<----- ; -----|
      |<----- , -----|
ROUTINGS --- VIA <node>--- TO <node> --- UNREACHABLE ---|
      |----- (<rout-info>)-|
  
```

This returns all routings for every neighbor node, showing for each neighbor a list of all destination nodes potentially reachable through that neighbor.

Using the "ROUTINGS VIA <node>" form of the ROUTINGS commands results in the response:

```

=> ROUTINGS VIA <node> --- TO <node> -- (<rout-info>) ---|
                                     |<----- ; -----|
                                     | - UNREACHABLE - |

```

This returns all routings for specified neighbor node, showing a list of all destination nodes potentially reachable through that neighbor.

Using the "ROUTINGS INUSE VIA ALL" form of the ROUTINGS command results in the response:

```

=>
|<----- ; -----|
|<----- , -----|
|
INUSE ROUTINGS ---VIA <node> ---TO <node> (<rout-info>) ---|

```

This returns in-use routing information for all neighbor nodes, showing the destinations currently reachable through each neighbor.

Using the "ROUTINGS INUSE VIA <node>" form of the ROUTINGS command results the response:

```

=> INUSE ROUTINGS VIA <node>--- TO <node> (<rout-info>) ---|
                                     |<----- , -----|

```

This returns in-use routing information for the specified neighbor node, showing the destinations currently reachable through that neighbor.

PROFILE SUMMARY AND DETAIL

These commands provide information about station profiles of a local node. There are two forms of the command, PROFILE, and PROFILE <profile-name>.

"PROFILE" returns a list of profile names.

"PROFILE <profile-name>" returns the list of station attributes and values contained in that profile.

Command Syntax

```
PROFILE -----|
-----|
      |-- <profile-name> --|
```

Response Syntax

The two forms of response follow:

Using the "PROFILE" form of the PROFILE command results in the response:

```

      |<----- , -----|
=> PROFILE --- ( --- <profile-name> --- ) -----|
```

This is a list of profile names currently known by the Station Level.

Using the "PROFILE <profile-name>" form of the PROFILE command results in the response:

```

      |<----- , -----|
=> PROFILE <prof-name> ( -- <sta-attr> = <value> -- ) -|
```

This is a list of station attributes associated with the profile.

STATION SUMMARY AND DETAIL

The STATION inquiry command provides various information about the stations. It can be used to provide station status information, station attribute information, and station COUNTS information. There are two major forms of the command: STATION and STATION <sta-name>.

The "STATION" form returns a list of stations with summary information about each. The "STATION <sta-name>" returns detailed information about the station selected.

Command Syntax

```
STATION -----|
---          | - <bdlc-station-name> -----|          |
              |                               |          |
              | - STATUS -                   |          |
              | -----                     |          |
              | - COUNTS -                   |          |
              | -----                     |          |
              | - DETAIL -                   |          |
              | -----                     |          |
              | - ALL -----                |          |
              | -----                     |          |
```

Response Syntax

The responses are described in the following subsections Station Summaries, and Station Detail.

STATION SUMMARIES

Using the "STATION" form of the STATION command results in the response:

```
=> STATION ----->
|<-----, -----|
|<bdlc-sta-n>:|---BDLC DEDICATED ---<status>-----| |
|             |-----|             |
| -BDLC SWITCHED ---| |-----+-----+|
| -BDLC SWITCHED ACU| | - WITH <neighbor> -|
```

where <bdlc-sta-n> = <bdlc-station-name>.

where <status> depends on the station type and is defined as:

```
----- ATTACHED -----|
|
|-- PENDING ATTACHED ---|
|-- CLOSED -----|
|-- PENDING CLOSED ----|
|-- WAITING -----|
|-- SAVED -----|
```

This provides a list of each station type, its status, and if it is connected to a neighbor, the neighbor identity.

STATION DETAIL

Using the "STATION <station-name>" form of the STATION command results in the response:

```

=> STATION <station> ( <-----, ----->
                        |
                        | <sta-attr & value> |
                        |
                        |
    
```

where <station> is defined as:

```

--- <bdlc-station-name> : --- BDL C DEDICATED -----|
                        |
                        | - BDL C SWITCHED -----|
                        |
                        | - BDL C SWITCHED ACU -----|
    
```

and where <station attribute> depends upon the attribute option selected: COUNTS, STATUS,DETAIL, OR ALL.

If attribute list option is *DETAIL*:

(the Station Level Manager Station Table attributes are selected)

HARDWARE	= <string>
STATIONTYPE	= BDL C DEDICATED/ BDL C SWITCHED/ BDL C SWITCHED ACU
NEIGHBORVALIDATE	= TRUE / FALSE
OUTGOINGENSEMBLE	= <ensemble-name>
INCOMINGENSEMBLE	= <ensemble-name>
PERMANENTENSEMBLE	= <ensemble-name>
NEIGHBOR	= <host-name> #<node-address>
AUTOINIT	= TRUE / FALSE
LINKSPEED	= <integer>
LINKEFFICIENCY	= <integer>
LINKMAXSEGMENTSIZ	= <integer>
MONITOR	= ON or OFF
LOCALSTATUS	= <enumerated value>
AUTOCOMMANDINPRO	= <enumerated value>
AUTOCOMMANDINPRO	= NULL/ ESTABLISHCALL/ESTABLISHCALLWITHNNA/ CLEAR CALL/ AWAITCALL

```

MANUALCOMMANDMODE      = TRUE / FALSE
MANUALCOMMANDINPROCESS = NULL/
                        OCPD/
                        OCPD <call-data>/
                        CCPD/
                        OSD/
                        CSD/
                        VALIDATEANDATTACH/
                        DETACH/
                        SENDTEST
PENDINGOPEN            = TRUE/ FALSE
REMOTEBUSYSTATUS      = BUSY/ NOT BUSY
SAVEMARKER            = TRUE/ FALSE

```

If the attribute list option = STATUS for BDLC

```

STATIONDIALOGSTATUS    } = OPEN/CLOSED/
CONNECTIONPORTDIALOGSTATUS } PENDING OPEN/
                             } PENDING CLOSED

STATIONMODE = OPEN/CLOSED
             PENDINGSENDSABM
             PENDINGRECEIVESABM
             PENDINGRESPONSEDISCONNECT
             PENDINGRESPONSE <integer>
             PENDINGSENDUA <integer>
             PENDINGSENDDM <integer>
             PENDINGSENDDISCONNECT <integer>

TESTSENDER = RESET/ PENDINGSEND/ PENDINGRESPONSE
TESTRECEIVER = RESET/ PENDINGSEND

```

If option selected = COUNTS for BDLC

```

FRAMERECEIVEDCOUNTER
FCSFAILURECOUNTER
FRAMESENTCOUNTER
IFRAMESENTCOUNTER

```

If the attribute option ALL is selected:

This returns a list of attributes and values which are those in Station Level Manager's station table entry for that station, plus the accessible attributes in the station itself.

ENSEMBLE SUMMARY AND DETAIL

This command provides information about the ensembles. There are two major forms of this inquiry command: ENSEMBLE and ENSEMBLE <ensemble-name>.

The "ENSEMBLE" form returns a list of ensemble names and types.

The "ENSEMBLE <ensemble-name>" form returns the type and station list for the specified ensemble.

Command Syntax

```
ENSEMBLE -----|
---          | -| <ensemble-name> -|
```

Response Syntax

The two forms of response are described:

Using the "ENSEMBLE" form of the ENSEMBLE command results in the response:

```
          |<-----, -----|
=> ENSEMBLE (--- <ens-name> : --- PERMANENT---) --|
          | - INCOMING -|
          | - OUTGOING -|
```

Using the "ENSEMBLE <ensemble-name>" form of the ENSEMBLE command results in the response:

```
=> ENSEMBLE <ens-name> --- INCOMING ----->
          | - OUTGOING --|
          | - PERMANENT -|
```

```
          |<-----, -----|
>-----| <bdlc-sta-name> -----|
```

CANDIDATE SUMMARY AND DETAIL

The CANDIDATE inquiry command provides information pertaining to the current port level candidates. There are two basic forms of the command: CANDIDATE and CANDIDATE <port-name>.

The "CANDIDATE" form returns a list of all candidates with summary information about each.

The "CANDIDATE <port-name>" form returns a list of all candidates with the specified port name with summary information about each.

Command Syntax

```
CANDIDATE -----|
--          |----- <port-name> -----|
```

Response Syntax

```
=>          |-----,-----|
          |-----,-----|
CANDIDATE--<port-name> -----(<match-info>-----)|
          |- [<subp-index>] -|
```

The subport index (<subp-index>) is returned only for local candidates.

candidate <match-info> for local candidates are:

```
USING PROCESS      = <string>
MY HOST            = <host-name>
YOUR HOST          = --- <host-name> ---
                   |--- *ANY -----|
MY NAME           = <string>
MY USERCODE       = <string>
YOUR NAME         = <string>
YOUR USERCODE     = <string>
PROGRAM           = <code-file-title> ON <family>
SECURITY          = --- PRIVATE -----
                   |--- PUBLIC -----|
                   |--- GUARDED BY <file-title> ---|

MATCHING STATUS   = --- NOT OFFERED -----
                   |--- OFFER OUTSTANDING ---|
                   |--- MATCH FOUND -----|
                   |--- TO BE CLOSED -----|
                   |--- AWAITING PLM DIALOG ---|
```

candidate <match-info> for remote candidates are:

```
MY HOST           = <host-name>
MY NAME           = <string>
MY USERCODE       = <string>
YOUR NAME         = <string>
PROGRAM           = <code-file-title> ON <family>

MATCHING STATUS   = --- INCOMING OFFER ---
                   |--- MATCH FOUND -----|
```

PORT COUNTS INQUIRY

This command provides information about number of users of the ports.

Command Syntax

```
PORTCOUNTS-----|  
-----
```

Response Syntax

```
      |<----- , -----|  
=> --- ALLOCATED PORTS ----- = <integer> ---|  
      |----- CANDIDATES ----- = <integer> -|  
      |----- REMOTE HOSTS ----- = <integer> -|
```

REMOTE HOSTS is returned only in Network mode.

MODE INQUIRY

This command provides information about the mode of the node, and the initialization file.

Command Syntax

```
NET -----|  
---
```

Response Syntax

```
=> --- NOT IN NETWORK MODE ----->  
|  
| - NETWORK |INITIALIZING--- ,INIT FILE = ---<file-title>--|  
| - NETWORK OPERATING --- | - *NULL -----|  
| - NODE IN SLOW SHUTDOWN|  
| - NODE SHUTTING DOWN --|  
  
>--- , NEXT INIT FILE = --- <file-title> -----|  
| --- *NULL -----|
```

SECTION 10

REPORTS

This section provides the syntax of the Network Services reports. These reports are originated by all levels of Network Services and presented to the Network Services Manager for display and logging (if required). This section is divided into four subsections: Network Services Manager Reports, Port Level Reports, Router Level Reports, and Station Level Reports.

As mentioned in Section 1, reports are normally sent to the ODT agent and by using the REPORTS + command, any agent requesting the reports. However, it should be noted that the agent requesting reports must have the authority to do so. This authority is granted by the Access Control command AUTHORIZE. For detail of this command, refer to Section 3.

NETWORK SERVICES MANAGER REPORTS

NETWORK SERVICES MANAGER PHASE CHANGE REPORT

This report is generated when there is a Network Services phase or mode change.

Report Syntax

```
NODE IS NOW ----- NETWORK INITIALIZING -----|
|
|  -- IN NETWORK MODE -----|
|  -- SHUTTING DOWN SLOWLY ----|
|  -- SHUTTING DOWN -----|
|  -- NOT IN NETWORK MODE -----|
```

MESSAGE FROM A REMOTE AGENT REPORT

This report is generated whenever a "TO" message is received from a remote agent. Receipt of this report only displays the message, it has no effect upon the node operation. All text within the message is treated for display purposes only.

Report Syntax

```
FROM <host-name> --- #<node-address> -- : -- <text> ---|
```

ROUTING REPORTS

This subsection provides details of the reports generated by the Router level. The reports include error reports, and reports about changes to network topology.

ROUTER FRAME ERRORS REPORTS

These are the error reports generated by the Router level. An illustration of the general syntax follows; the syntax error name describes the actual error:

Report Syntax

```
ERROR IN MESSAGE --- FROM <host-name> #<node-address> ----->
>----- TO <host-name> #<node-address> : ---->
>--- FRAME UNDELIVERABLE -----|
|
| - TRANSIT COUNT OVERFLOW -----|
| - MAX SEGMENT SIZE <intgr> EXCEEDS <intgr> -|
| - FRAME TOO SHORT -----|
| - INVALID FRAME FORMAT, TYPE = <hex-number>-|
| - UNKNOWN FRAME FORMAT, TYPE = <hex-number>-|
|
| v v
```

```
V
- INVALID DESTINATION : <node-address> -----V
- UNKNOWN ORIGINATOR : <node-address> -----
- INVALID NETCHANGE SUBJECT: ---|
  |<-----|
  | - <node-address> FROM <node-address> -----
- INVALID LINKCHANGE ORIGINATOR: <node-addr>
- INCONSISTENT MRF = <integer> ---|
  |<-----|
  | - & MHC = <integer> FROM <node-addr> --
- UNREACHABLE DESTINATION: <node-addr>--|
  |<-----|
  | - FOR TRACE RESULT FROM <node-address>--
```

ROUTING TRACE REPORT

A trace operation is initiated in response to a START TRACE command. Subsequently reports are generated by all nodes along a trace path, and are then delivered to the originating node in the following form:

Report Syntax

```

ROUTING TRACE FROM <node> : <----- , ----->
                             --- REFERENCE NUMBER = <integer> ---
                             - TRANSIT COUNT    = <integer> -----
                             - SOURCE = <host-name> #<node-address>--
                             - DESTINATION= <host-n> #<node-address>
                             - MY RESISTANCE FACTOR = <integer>-----
                             --- REACHABLE -----
                             | - UNREACHABLE -- |
                             - HOP COUNT ----- = <integer> -----
                             - PATH RESISTANCE FACTOR =<integer> ---
                             - NEIGHBOR = <host-name>#<node-address>
                             - LINK RESISTANCE FACTOR =<integer> ---
    
```

Where <node> is the node which sent the report.
Where <host-n> = <host-name>.

DESTINATION NODE STATUS CHANGE REPORT

This report is generated by the Router whenever its routing parameters change for a remote node.

Report Syntax

ROUTING STATUS TO <host-name> #<node-address> CHANGED : ---->

```

|<-----, -----|
>----- REACHABLE -----|
| - * - | | - UNREACHABLE - |
| - MAX SEGMENT SIZE = <integer> ---- |
| - VIA <host-name> #<node-address> - |
```

All parameters are returned whether changed or not. The "*" indicates that a parameter has been changed.

STATION LEVEL REPORTS

This subsection provides details on the Station level reports. These reports include error reports and status reports that are generated by the Station level. The reports are detailed under two categories: Neighbor Status Reports and Station Reports.

The Neighbor Status Reports provide information that concerns the logical connection to the neighbor. The Station Reports provide information concerning the individual physical links.

NEIGHBOR STATUS REPORTS

In these reports <neighbor> is <host-name> #<node address>.

Report Syntax

<neighbor> RESTARTING -----|

This report is generated whenever a GREETINGS 1 frame is received indicating that the neighbor is initializing, and the neighbor was thought to be attached; that is, it must be RESTARTING.

Report Syntax

<neighbor> DISCONNECTED, NEIGHBOR BUSY TIMEOUT EXCEEDED

This report indicates that all connections to the neighbor were in a remote busy state (no activity for more than NEIGHBOR BUSY TIMEOUT), and have now been closed.

STATION REPORTS

Validation Failure Reports

Reports are generated by the Station level whenever a validation failure is detected. That is, a call was made, or received by a station, and it could not complete the validation process, either because of security or incompatibility restrictions.

This may require operator intervention or it may be a correct rejection of an invalid link attempt.

If operator intervention is required, it would be the operators task to determine the cause of the validation failure, and correct it. Most likely causes would be incorrect NODEPASSWORDs, incompatible software (VERSION information), invalid Network Maximum Segment Size, or unknown node address (if NEIGHBOR VALIDATE true).

Report Syntax

```
--- LOCAL ---- VALIDATION TO <neighbor> FAILED ON <station-name>  
| - REMOTE - |
```

Station Attach and Detach Report

These reports are generated whenever the attachment of a station is changed. Attachment occurs when the validation process completes, and the station is attached to the router queues for traffic. Detachment occurs when the station is no longer attached to the router for traffic.

Report Syntax

```
STATION <sta-name> --- ATTACHED TO ----- <neighbor> -----|  
| - DETACHED FROM- |
```

Station Failure Report

Station failure reports provide information on station failures that occur during the station functions: Open Connection Port Dialog, Open Station Dialog, or Send Test. Each failure is reported and the <reason> field describes the failure type.

Report Syntax

```
STATION <sta-name> FAILED TO <action> BECAUSE <reason>
```

The <action> is one of the following:

- . OPEN CONNECTION PORT DIALOG
- . OPEN STATION DIALOG
- . SEND TEST

Test Reports

These reports are caused by the receipt of Test messages by a station. Test RECEIVED indicates that a remote station sent a Test message, and provides the text of that message. Test ECHOED indicates that the Test message sent by this station was echoed back by the remote station. No checking of the message is made by the station, it is just displayed as received.

Report Syntax

```
TEST --- RECEIVED --- FROM <neighbor> -----|
      | - ECHOED --- |                       | - : <text> -- |
```

Station Error Report

These reports are generated whenever the station detects an error situation. The error description is provided in the <station-error-message> of the report.

Report Syntax

```
STATION -- <sta-name> -- ERROR: <station-error-message>
```

PORT LEVEL REPORTS

Reports that are generated by the Port Level Manager are presented in this subsection.

HOST SAVED REPORT

Report Syntax

HOST SAVED: <host-name> #<node-address>

This report occurs when a host actually becomes saved. That is, after a SAVE HOST command and all subport communication is stopped.

PORT LEVEL ERROR REPORTS

Validation Failure Report

This report is generated whenever a validation failure is detected. This can occur legitimately when an unacceptable host attempts to open a dialogue, or it may occur due to some operational error. If this occurs, the most likely causes are: an invalid password, an invalid node address, or an invalid host name, or it may be that the HOST VALIDATION is on inadvertently. Operator intervention is required to overcome this, should it be desired.

Report Syntax

HOST VALIDATION FAILURE ON <host-name> #<node-address>

Subport Dialog Establishment Failure Report

This report occurs whenever a subport dialogue management error is detected. This should not normally occur. It indicates a logical failure in the port mechanism, or that port frames are being received corrupted.

Report Syntax

PLM SUBPORT DIALOG ESTABLISHMENT FAILURE

Port Level Error Report

Errors detected in messages, received by the Port level are reported with this report. The error type is described in the report.

Report Syntax

ERROR IN MESSAGE FROM <host-name> #<node-address>: ----->

```

>---- UNKNOWN PORT ADDRESS: <port-address> -----|
|
| - INVALID FRAME FORMAT, TYPE = <hex-number> -
| - INVALID SUBPORT ADDRESS:-----|
|
| <-----|
|
| -<subport-addr> FOR PORT <port-addr>-|

```

APPENDIX A

OIM MESSAGE SYNTAX

RAILROAD DIAGRAMS

In this document the syntax of each OIM is presented in the form of railroad diagrams. A railroad diagram is a technique used to graphically represent the syntax of language elements. Traversing a railroad diagram from left to right or in the direction of the arrow heads, and adhering to the limits illustrated by bridges produces a syntactically valid statement.

Continuation from one line of a diagram to another is represented by a right arrow > appearing at the end of the current line and the beginning of the next line.

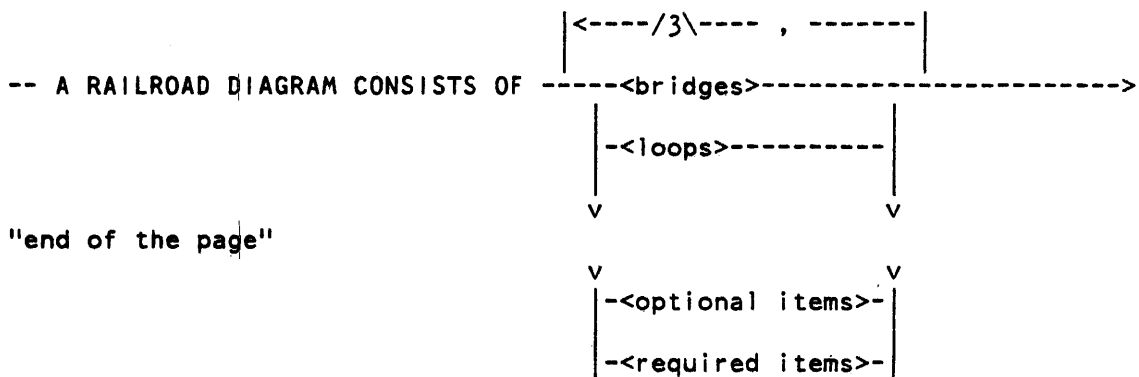
Continuation from one page to another is avoided wherever possible. However, where unavoidable, the use of the right arrow >, and for the case of items listed vertically, the inverted arrow v, are used to denote continuation.

The syntax diagram is terminated by a vertical bar |.

Items contained in broken brackets < > are syntactic variables which are further defined in the appendix, or require the user to supply the requested information.

Upper case items must appear literally, but where abbreviations can be used, the abbreviated items are underlined>.

Example



>- AND IS TERMINATED BY A VERTICAL BAR. -----|

The following syntactically valid statements may be constructed from the above diagram:

A RAILROAD DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items>, <required items>, <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

RAILROAD COMPONENTS

<required items>

No alternate path through the railroad diagram exists for required items.

Example

-- REQUIRED ITEM -----|

<optional items>

Items shown as a vertical list indicate that the user must make a choice of the items specified. An empty path through the list allows the <optional item> to be absent.

Example

-- REQUIRED ITEM -----|
| --<optional item-1>- |
| --<optional item-2>- |

The following valid statements may be constructed from the previous diagram:

```
REQUIRED ITEM
REQUIRED ITEM <optional item-1>
REQUIRED ITEM <optional item-2>
```

<loop>

A <loop> is a recurrent path through a railroad diagram. A <loop> must be traversed in the direction of the arrow heads, and if any limits are imposed, they are specified by bridges. A <loop> has the following general format:

```
|<--- <bridge> ---- <return character> -|
-----<object of the loop>-----|
```

Example

```
|<-/1\----- , -----|
-----<optional item-1>-----|
|-----<optional item-2>-----|
```

The following statements can be constructed from the previous diagram:

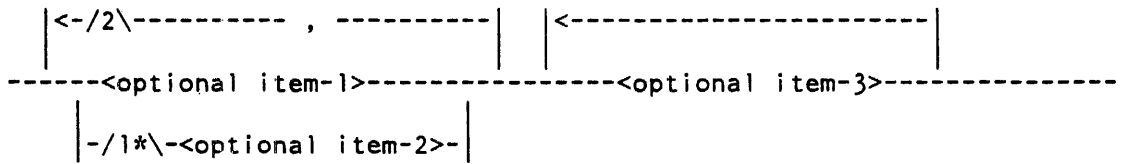
```
<optional item-1>
<optional item-2>
<optional item-1>,<optional item-1>
<optional item-2>,<optional item-1>
```

<bridges>

A <bridge> illustrates the minimum or maximum number of times a path may be traversed in a railroad diagram. The forms of <bridges> are:

```
/n\ n is an integer which specifies the maximum number of
times the path may be traversed.
/n*\ n is an integer which specifies the minimum number of
times the path may be traversed.
```


Example



The first loop may be traversed a maximum of two times, however, the path for <optional item-2> must be traversed at least one time. The second loop can be traversed any number of times, that is, <optional item-3> can be entered any number of times.

The following statements can be constructed from the previous diagram.

```

    <optional item-1>,<optional item-2>,<optional item-3>
    <optional item-2>,<optional item-2>,<optional item-3>
    <optional item-2>,<optional item 1>,<optional item-3>
    <optional item-2>,<optional item-3>,<optional item-3>
  
```

Words appearing in all upper case are keywords and must appear as presented. Words in lower case and embraced in wedges are symbols for user-supplied words or values. Where keywords can be abbreviated, the abbreviation is denoted by the letters underlined in the keywords. Each keyword, keyword abbreviation, or user-supplied item must be delimited from another such word by at least a single space or by a special character (for example, parentheses, comma, period, colon) defined in the syntax. Redundant spaces are ignored.

DEFINITION OF COMMON SYNTACTIC ELEMENTS

<letter> ::= (A thru Z, a thru z)

<digit> ::= (0 thru 9)

```

    |<-----|
    |-----<digit>-----|
    <integer> ::= -----|
  
```

<character> ::= (any EBCDIC graphic character except quote,
but including quote-image)

<separator> ::= (hyphen or underline)

<station-name> ::= <name>

<host-name> ::= <name>

<bdlc-station-name> ::= <name>

```

      |<-----|
<name> ::= -----|
      |<letter>-----|
      | - <digit> --- | | - <separator> - |
  
```

note:

- . upper and lower case letters are equivalent.
- . the separators are equivalent to each other.
- . names are not distinguished beyond the 17th character.

```

      |<-- <digit> ---|
<value> ::= -----|
      |<digit>-----|
      |-----|
      | *DEFAULT -----|
      |-----|
  
```

- . If *DEFAULT is used for <value>, the default value is selected.

<response indicator> ::= =>

- . This indicator signifies that response syntax follows.

<quote> ::= (the double quote character, ")

<quote-image> ::= <quote> <quote>

```

<string> ::= -----|
      |<name>-----|
      |<-----|
      |<quote> --- <character> --- <quote> -|
      |-----|
      | *NULL -----|
      |-----|
  
```

notes:

- . if a <quote> is to be entered as a <character> it is replaced by <quote-image>.
- . arbitrary character strings are embraced in quotes to prevent syntactic confusion.
- . *NULL is used to define an empty string. Empty strings are displayed in reports and responses.

The following elements are syntactically equivalent to <name>

<code><host-name></code>	<code><port-name></code>	<code><station-name></code>
<code><profile-name></code>	<code><ensemble-name></code>	<code><program-agent-name></code>

`<text> ::=` any sequence of EBCDIC characters except those used for textual control (for example, on ODTs the continue mark, on files the end-of-command mark).

`<node-address> ::= <integer>`

`<file-title> ::=` ----- <name> ----->

- (<usercode>) -
----- * -----

<-----

--/11\-- <slash> <name> -	- ON <name> -
---------------------------	---------------

`<neighbor> ::=` --- <host-name> -----|

- #<node address> -----
- <host-name> #<node-address> -

KEY WORD REPRESENTATION

In this document, key words used for commands, inquiries, attributes, and so on, are represented in two ways:

The first, and syntactically correct, is the form of contiguous, upper case, alphabetic characters,

for example: PORTSEGMENTTIMEOUT.

The other, not syntactically correct, is the separation of the component words for readability,

for example: PORT SEGMENT TIMEOUT.

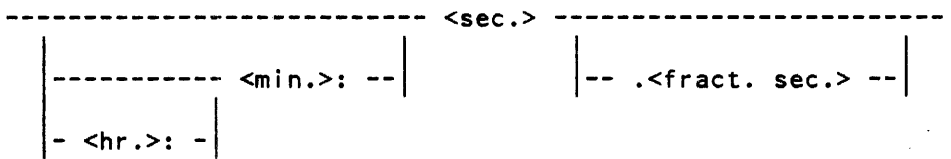
The non-syntactically correct representations are only used in textual descriptions. Error messages, examples, and so on, all use the precise syntax.

Where all characters of a keyword are not required, those required are underlined. E.g., PORTSEGMENTTIMEOUT, depicted underlined below, can be entered in the abbreviated form of PSTO.

PSTO is equivalent to PORTSEGMENTTIMEOUT

TIME INTERVALS

Throughout this document, whenever <time-interval> is specified, the following syntax applies:



where <hr.>, <min.>, <sec.>, and <fract. sec.> are integers.

Each time-interval attribute can have a different granularity of time. The units of granularity are specified, in this document, by implication of maximum-range-units for each command attribute.

For example, the PORTSEGMENTTIMEOUT command shows the range maximum as 65535 secs.; this attribute is granular to the second. If the value supplied is 9:5.7 (9 minutes, 5.7 seconds) then the value used internally is 9 minutes, 6 seconds. For example:

```

the command:
    PORTSEGMENTTIMEOUT = 9:5.7

yields the response:
    PORTSEGMENTTIMEOUT = 9:6
    
```

Some useful conversion values:

```

65535 seconds = 18:12:15 (hrs: mins.: secs.)
1440 minutes  = 24:00:00 = 1 day
3600 seconds  = 1:00:00 = 1 hour.
    
```

BDLC CALL DATA <dblc-call-data>

In this document <bdlc-call-data> is defined as:

```
----- " <string> "-----|
```

The <string> is a phone number to be used by an ACU.

It must appear between quotes.

It must be a string of characters composed of decimal digits, asterisks (*), number symbol (#), hyphens (-), End of Number characters (E), and Separators (S).

The decimal digits correspond to HEX digits, the asterisks correspond to HEX A, the number symbol corresponds to HEX B, the End of Number character corresponds to HEX C, and the Separator corresponds to HEX D. The hyphens are ignored.

These characters are used to perform ACU dial operations. Digits specify the telephone number to dial. Some dialing conventions require an end of number (EON) to define the end of the dial operation; the End of Number (E) is used for this.

Where delays between dial digits is required, such as internal telephone systems to obtain an outside line, the Separator (SEP) character (S) provides a one-digit delay time. On some ACUs the Separator character provides a dialing delay but does not put any output on the line, and the delay may be for one digit time or some ACU-defined time. In either case the delay is used to allow the next dial tone to be present, or provide delays required for reliable dialing.

A maximum of 20 characters, not counting hyphens, can appear.

Examples

To dial the following telephone number (215) 648 4000
<bdlc-call-data> is:

```
"2156484000" or "215-648-4000"
```

To dial the same number from an internal telephone system that requires a second dialing tone. That is, a delay of one digit time is needed after the first digit. The <bdlc-call-data> is:

```
"9S2156484000" or "9-S-215-648-4000"
```

Where the "S" introduces a one digit delay.

Delays can be introduced anywhere within the string to provide delays suited for the telephone system, and dialing procedures.

APPENDIX B

HOW TO PICK MAX HOP COUNT AND MAX RESISTANCE FACTOR (MHC AND MRF)

The values of MRF and MHC are constant within a particular network. The selection of these values for a particular network is important and may be non-trivial for some networks. This appendix describes what the "best" values of MRF and MHC are and gives some idea of their bounds.

USE OF MRF AND MHC IN THE ALGORITHM

MRF and MHC are used by the Router in the BNA network to determine reachability of a destination node. If a NETCHANGE message is received with either (resistance factor + linkrf \geq MRF) or (hop-count + 1 \geq MHC), the subject node is marked as unreachable over the link on which the NETCHANGE was received.

THE "BEST" SIZE FOR MHC AND MRF

The "best" size may have different meanings for different people, but one definition is the following:

MHC and MRF are the smallest value such that any possible physical path in the network (without entering a node twice) between two nodes has (hop-count < MHC) and (total resistance-factor < MRF).

COST OF MHC AND MRF BEING TOO LARGE

If MHC and/or MRF are too large (larger than the "best" value), more NETCHANGE messages may be exchanged among the nodes than with the "best" value. How many more depends on how "too large" the values are and upon the network topology.

COST OF MHC AND MRF BEING TOO SMALL

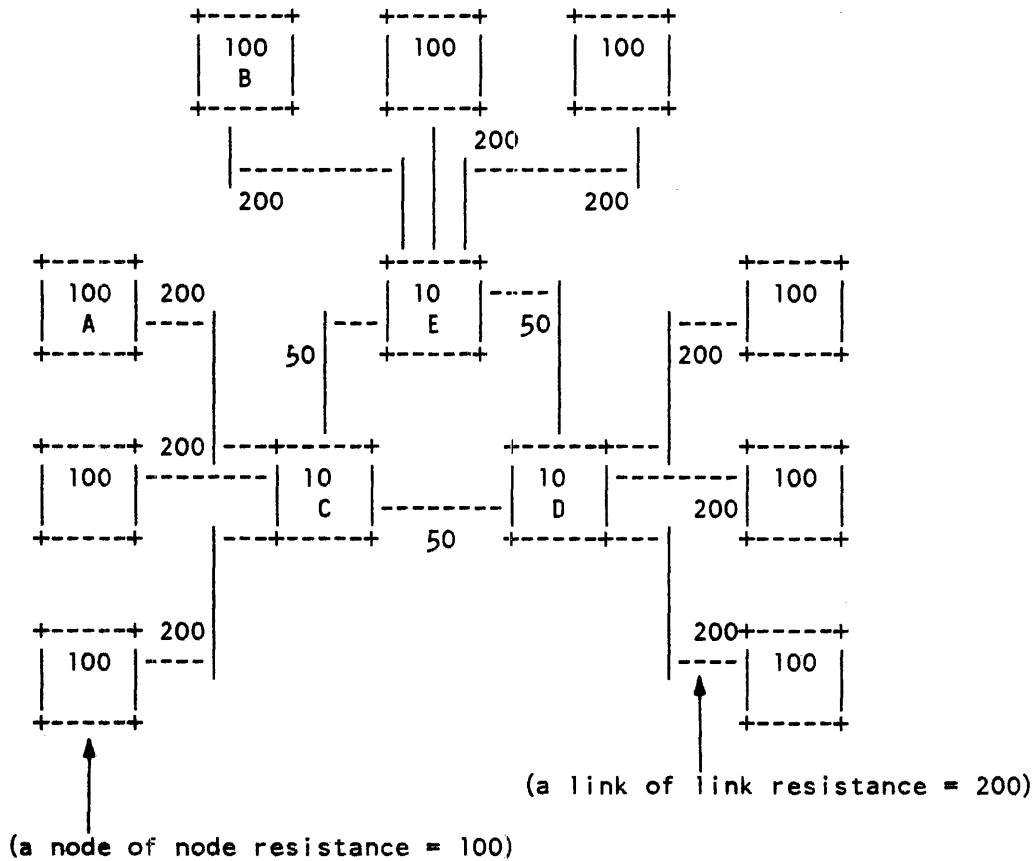
If MHC and/or MRF are too small, (smaller than the "best" value), artificial conditions of unreachability are introduced, that is, there exists a physical path between two nodes, but both of the nodes believe that it does not exist. (NOTE: this may be considered a feature by some network designers. For example, although there are physical paths of up to 20 hops, the network designer may not want any dialogues to be established with a hop-count of more than 8.)

BOUNDS ON MHC AND MRF FOR THE "BEST" VALUE

Clearly, the smallest either MHC or MRF can be is 1. Given a network with N nodes, the most that MHC can be is N, because a path may not go through a node twice. An upper bound on MRF is the sum of the highest N-2 node resistances, plus the highest N-1 link resistances in the network. However in many cases the upper bound is not the "best" value, particularly MRF (see example.)

The process of selecting the "best" MHC is roughly equivalent to the process of finding a path which goes through each node once and only once. The process for determining MRF is more difficult. Both problems can only be solved completely by an exhaustive search of all possibilities.

Example :



Upper bound MHC = 12; actual "best" value = 5

Upper bound MRF = 2810; actual "best" value = 531

The upper bound of MHC is the number of nodes in this network. It is certainly not a good choice; there is no possibility of taking 11 hops from any node to any other node in this network.

The actual "best" value is chosen because the maximum number of hops taken between any two nodes is four. For example: If the link between C and E is broken, the number of hops between A and B becomes 4 (A to C, C to D, D to E, and E to B). Thus the "best" choice for MHC is 5 (4 + 1).

Of course 5 need not necessarily be the value chosen, as there may be other factors to be considered. There are cases where a lower hop count would suffice, such as if all nodes are not required to communicate with each other. Also, in the example shown above, if the broken link condition could not occur (very unlikely, however), then a MHC of 4 could be used.

The choices for MRF are made in a similar manner.

APPENDIX C

GENERAL INDEX, ALPHABETICAL ORDER

This index includes Commands, Reports, Error responses, and Station attributes.

ACURSPTIMER	5 - 13
ADD CONNECTION	5 - 26
ADD ENSEMBLE	5 - 22
ADD HOST	5 - 3
ADD NODE	5 - 4
ADD PROFILE	5 - 16
ADD STATION	5 - 18
ADDED STATION	5 - 25
ALL REQUIRED ATTRIBUTES NOT SET	5 - 19
ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET	2 - 4
ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET	1 - 15
ALL STATIONS IN ENSEMBLE ARE CLOSED	6 - 4
AT	1 - 6
ATTRIBUTE INCONSISTENT WITH STATION TYPE.	5 - 19
AUTHORITY CANNOT BE GRANTED OR REMOVED.	3 - 6
AUTHORIZE	3 - 4
AUTOINIT	5 - 8
AWAITCALL BY	6 - 1
BDLC FAILURE ON <station>	7 - 3
CALL DATA NOT REQUIRED.	6 - 7
CALL DATA REQUIRED.	6 - 7
CANDIDATE	9 - 18
CHECKPOINTTIMER	5 - 12
CLEAR HOST	6 - 10
CLEARCALL	6 - 3
CLOSE CPD or SD	7 - 1
COMMAND OUT OF SEQUENCE	7 - 3
CONNECTION IN USE	5 - 30
CONNECTION IN USE	5 - 29
CONTINUE	8 - 4
CPDMONITOR	5 - 12
CTSRPTIMER	5 - 10
DATARATESELECTOR	5 - 9
DCDOFFDELAYTIMER	5 - 10
DEDICATED	1 - 11
DELETE CONNECTION	5 - 29
DELETE ENSEMBLE	5 - 24
DELETE HOST	5 - 5
DELETE NODE	5 - 6
DELETE PROFILE	5 - 17
DELETE STATION	5 - 20

DELETED STATION NOT IN ENSEMBLE	5 - 25
DETACH	7 - 1
DIALRESPONSETIMER	5 - 11
DIALRETRYCOUNT	5 - 13
DISCARD	8 - 5
<neighbor> DISCONNECTED, NEIGHBOR BUSY TIMEOUT EXCEEDED	10 - 6
DPRDELAYTIMER	5 - 13
DSRRSPTIMER	5 - 11
DUPLICATE AGENT NAME.	3 - 11
EFFICIENCY	5 - 8
ENDINITIALIZATION	2 - 4
ENSEMBLE	9 - 17
ENSEMBLE ALREADY ADDED.	5 - 23
ENSEMBLE IN USE	5 - 24
ERROR IN MESSAGE	10 - 2
ERROR IN MESSAGE FROM	10 - 10
ERROR	7 - 3
ESTABLISHCALL	6 - 4
FCSFAILURECOUNTER	5 - 12
FILE ALREADY SUSPENDED.	8 - 3
FILE LOADING IN PROCESS	8 - 2
FRAMERECEIVEDCOUNTER.	5 - 12
FRAMESENTCOUNTER.	5 - 12
FROM	10 - 2
FRSPTIMERLONG	5 - 11
FRSTIMERSHORT	5 - 11
HARDWARE	5 - 8
HN	1 - 5
HOST	9 - 1
HOST ALREADY ADDED.	5 - 4
HOST NAME	5 - 5
HOST NAME	5 - 3
HOST NAME	5 - 7
HOST NAME DOES NOT MATCH.	4 - 2
HOST NOT DELETED.	5 - 5
HOST SAVED.	10 - 9
HOST VALIDATION FAILURE ON	10 - 9
HOSTINACTIVEDISCONNECT	4 - 21
HOSTINACTIVETIMEOUT	4 - 21
HOSTUNREACHABLETIMEOUT	4 - 22
IFRAMESENTCOUNTER	5 - 13
ILLEGAL PASSWORD.	3 - 3
IMPROPER ENSEMBLE KIND.	5 - 28
IMPROPER ENSEMBLE TYPE.	6 - 2
IMPROPER ENSEMBLE TYPE.	6 - 7
IMPROPER FILE FORMAT.	8 - 2
IMPROPER PARAMETER.	5 - 30
IMPROPER PHASE.	1 - 15
IMPROPER SETTING OF STATION ATTRIBUTE	5 - 19
IMPROPER STATION KIND	5 - 28

IMPROPER STATION TYPE	6 - 2
IMPROPER STATION TYPE	6 - 7
INCORRECT PARAMETERS.	5 - 28
INVALID AUTHORIZATION OPTION.	3 - 6
INVALID HOST INACTIVE TIMEOUT	4 - 23
INVALID HOST INACTIVE DISCONNECT OPTION	4 - 23
INVALID HOST NAME	5 - 5
INVALID HOST NAME	5 - 7
INVALID HOST UNREACHABLE TIMEOUT.	4 - 23
INVALID LINK RESISTANCE FACTOR.	4 - 17
INVALID LOGGING OPTION.	4 - 8
INVALID MAX HOP COUNT	4 - 15
INVALID MAX PROGRAM AGENT VALUE	3 - 9
INVALID MAX RESISTANCE FACTOR	4 - 15
INVALID MONITOR OPTION.	4 - 11
INVALID MONITOR TIME INTERVAL	4 - 11
INVALID NEIGHBOR BUSY TIMEOUT VALUE	4 - 25
INVALID NEIGHBOR GREETING TIMEOUT VALUE	4 - 25
INVALID NEIGHBOR RESTART TIMEOUT VALUE.	4 - 15
INVALID NETWORK MAX SEGMENT SIZE.	4 - 6
INVALID NODE ADDRESS.	4 - 2
INVALID NODE ADDRESS.	5 - 3
INVALID NODE ADDRESS.	5 - 6
INVALID NODE ADDRESS.	5 - 4
INVALID NODE ADDRESS.	5 - 7
INVALID NODE ADDRESS.	1 - 15
INVALID NODE RESISTANCE FACTOR.	4 - 15
INVALID NODE UP TIMEOUT VALUE	4 - 15
INVALID PORT COMPRESSION ALLOWED OPTION	4 - 20
INVALID PORT LOG TIME INTERVAL.	4 - 12
INVALID PORT RESUME READY FACTOR.	4 - 20
INVALID PORT RETRY LIMIT.	4 - 20
INVALID PORT SEGMENT TIMEOUT INTERVAL	4 - 20
INVALID PORT WINDOW SIZE.	4 - 20
INVALID PROFILE REFERENCE	5 - 19
INVALID PROGRAM AGENT SECURITY OPTION	3 - 9
INVALID REMOTE HOST	3 - 11
INVALID SLM PHASE	7 - 3
INVALID STATION ATTRIBUTE VALUE	5 - 16
INVALID STATION ATTRIBUTE NAME.	5 - 16
INVALID STATION NAME.	6 - 9
INVALID STATION NAME.	6 - 8
INVALID STATION PHASE ON	7 - 3
INVALID STATION	5 - 20
INVALID TRACE DESTINATION	6 - 13
INVALID TRACE SOURCE.	6 - 13
INVALID USERCODE.	3 - 6
LAST REMAINING STATION MAY NOT BE DELETED	5 - 25
LINKRESISTANCE	4 - 16
LOAD	8 - 1

LOCALIDENTITY	4 - 1
LOCAL VALIDATION TO <neighbor> FAILED ON <station>.	10 - 7
LOGCOMMENT	6 - 12
LOGGING	4 - 7
MAX SEGMENT SIZE	5 - 9
MAXHOPCOUNT	4 - 12
MAXPROGRAMAGENTS	3 - 8
MAXRESISTANCEFACTOR	4 - 12
MESSAGE TEXT EXCEEDS MAX SEGMENT SIZE	6 - 14
MISSING NODE ADDRESS.	5 - 4
MODIFY CONNECTION	5 - 30
MODIFY ENSEMBLE	5 - 25
MODIFY STATION	5 - 21
MONITOR	5 - 9
MONITOR	4 - 9
NEIGHBOR	9 - 4
NEIGHBOR IS NOT A NODE.	5 - 28
NEIGHBORBUSYTIMEOUT	4 - 24
NEIGHBORGREETINGTIMEOUT	4 - 24
NEIGHBORRESTARTTIMEOUT	4 - 13
NET	9 - 21
NET	2 - 1
NETWORKMAXSEGMENTSIZ	4 - 5
NETWORKVERSION	4 - 5
NO CALLS WITH NEIGHBOR EXIST.	6 - 4
NO FILE BEING LOADED.	8 - 3
NO MORE THAN TWO PROFILES MAY BE SPECIFIED.	5 - 19
NO STATION AVAILABLE.	6 - 2
NO STATION AVAILABLE.	6 - 7
NO SUCH <entity>	1 - 15
NO SUCH CONNECTION.	5 - 30
NO SUCH CONNECTION.	5 - 29
NO SUCH ENSEMBLE.	5 - 24
NO SUCH ENSEMBLE.	5 - 25
NO SUCH ENSEMBLE.	6 - 2
NO SUCH ENSEMBLE.	6 - 4
NO SUCH ENSEMBLE.	6 - 7
NO SUCH HOST OR NODE.	6 - 14
NO SUCH HOST OR NODE.	3 - 3
NO SUCH HOST.	3 - 11
NO SUCH REMOTE HOST	6 - 11
NO SUCH REMOTE HOST	6 - 10
NO SUCH REMOTE HOST	6 - 11
NO SUCH STATION GROUP	6 - 2
NO SUCH STATION	6 - 2
NO SUCH STATION	6 - 7
NO SUCH STATION	6 - 4
NO SUCH USERCODE.	3 - 11
NO SUSPENDED FILE	8 - 5
NO SUSPENDED FILE	8 - 4
NODE	9 - 4
NODE ALREADY ADDED.	5 - 4

NODE ALREADY DELETED.	5 - 7
NODE IS NOW	10 - 1
NODE NOT DELETED.	5 - 7
NODE NOT DELETED.	5 - 7
NODE UNREACHABLE.	6 - 14
NODEPASSWORD	3 - 1
NODERESISTANCEFACTOR	4 - 13
NODESUMMARY	4 - 26
NODEUPTIMEOUT	4 - 14
NOT DELETED	5 - 20
NOT DELETED	5 - 20
NOT DELETED	5 - 20
NOT IN NETWORK MODE	1 - 14
<file-title> NOT PRESENT FOR LOADING.	8 - 2
NW	1 - 10
OPEN CPD or SD	7 - 1
PLM SUBPORT DIALOG ESTABLISHMENT FAILURE.	10 - 9
PORTCOMPRESSIONALLOWED	4 - 17
PORTCOUNTS.	9 - 20
PORTLEVELINCARNATION	4 - 3
PORTLOGINTERVAL	4 - 11
PORTRESUMEREADY	4 - 18
PORTRETRYLIMIT	4 - 18
PORTSEGMENTTIMEOUT	4 - 18
PORTWINDOWSIZE	4 - 18
PROFILE	9 - 12
PROFILE ALREADY DEFINED	5 - 16
PROFILE NOT DEFINED	5 - 17
PROGRAM	3 - 9
PROGRAMAGENTSECURITY	3 - 8
READY HOST	6 - 10
READY STATION	6 - 8
RELEASE	1 - 16
REMOTEBUSYMONITOR	5 - 13
REMOTE VALIDATION TO <neighbor> FAILED ON <station>	10 - 7
REPORTS	1 - 16
<neighbor> RESTARTING	10 - 6
RETRYDELAYTIMER	5 - 14
RETRYLIMITCHECKPT	5 - 9
RETRYLIMITSABMDIS	5 - 10
ROUTER VALIDATE IS FALSE.	5 - 4
ROUTERHEADERSIZE	4 - 3
ROUTING STATUS TO	10 - 5
ROUTING TRACE FROM	10 - 4
ROUTINGREFRESH.	6 - 12
ROUTINGS	9 - 8
SAVE HOST	6 - 11
SAVE STATION	6 - 9
SELECTSTANDBY	5 - 10
SENDTEST	7 - 1
SHUTDOWN CANNOT BE CANCELLED.	2 - 3
SPEED	5 - 9

STARTTRACE	6 - 13
STATION	10- 7
STATION	10- 8
STATION	10- 8
STATION	9 - 13
STATION ALREADY CLOSED.	6 - 4
STATION ATTRIBUTE MAY NOT BE SET.	5 - 21
STATION FAILURE	6 - 3
STATION FAILURE	6 - 7
STATION NOT AVAILABLE	6 - 7
STATION NOT AVAILABLE	6 - 2
STATION NOT CLOSED.	5 - 21
STATIONS IN ENSEMBLE NOT OF PROPER TYPE	5 - 23
STATIONS IN ENSEMBLE HAVE NOT BEEN ADDED.	5 - 23
STOP	8 - 3
TEST	10- 8
TO	6 - 14
TRANSITCOUNTLIMIT	4 - 4
TUNINGS	4 - 25
UNAUTHORIZED AGENT.	1 - 14
UNREACHABLE TRACE DESTINATION	6 - 13
UNREACHABLE TRACE SOURCE	6 - 13
USERCODE MAY NOT BE SPECIFIED	3 - 11
VALIDATE	3 - 6
VALIDATE OPTION INVALID	3 - 7
VALIDATEANDATTACH	7 - 1
VALIDATION FAILURE ON	7 - 3
VALIDATION FAILURE.	6 - 7
VALIDATION FAILURE.	6 - 2
VERSION	4 - 4
WINDOWSIZE	5 - 10

APPENDIX D

COMMAND INDEX, ALPHABETICAL ORDER

ADD CONNECTION	5 - 26
ADD ENSEMBLE	5 - 22
ADD HOST	5 - 3
ADD NODE	5 - 4
ADD PROFILE	5 - 16
ADD STATION	5 - 18
AT	1 - 6
AUTHORIZE	3 - 4
AWAITCALL BY	6 - 1
CANDIDATE	9 - 18
CLEAR HOST	6 - 10
CLEARCALL	6 - 3
CLOSE CPD or SD	7 - 1
CONTINUE	8 - 4
DEDICATED	1 - 11
DELETE CONNECTION	5 - 29
DELETE ENSEMBLE	5 - 24
DELETE HOST	5 - 5
DELETE NODE	5 - 6
DELETE PROFILE	5 - 17
DELETE STATION	5 - 20
DETACH	7 - 1
DISCARD	8 - 5
ENDINITIALIZATION	2 - 4
ENSEMBLE	9 - 17
ESTABLISHCALL	6 - 4
HN	1 - 5
HOST	9 - 1
HOSTINACTIVEDISCONNECT	4 - 21
HOSTINACTIVETIMEOUT	4 - 21
HOSTUNREACHABLETIMEOUT	4 - 22
LINKRESISTANCE	4 - 16
LOAD	8 - 1
LOCALIDENTITY	4 - 1
LOGCOMMENT	6 - 12
LOGGING	4 - 7
MAXHOPCOUNT	4 - 12
MAXPROGRAMAGENTS	3 - 8
MAXRESISTANCEFACTOR	4 - 12
MODIFY CONNECTION	5 - 30

MODIFY ENSEMBLE	5 - 25
MODIFY STATION	5 - 21
MONITOR	4 - 9
NEIGHBOR	9 - 4
NEIGHBORBUSYTIMEOUT	4 - 24
NEIGHBORGREETINGTIMEOUT	4 - 24
NEIGHBORRESTARTTIMEOUT	4 - 13
NET	9 - 21
NET	2 - 1
NETWORKMAXSEGMENTSIZ	4 - 5
NETWORKVERSION	4 - 5
NODE	9 - 4
NODEPASSWORD	3 - 1
NODERESISTANCEFACTOR	4 - 13
NODESUMMARY	4 - 26
NODEUPTIMEOUT	4 - 14
NW	1 - 10
OPEN CPD or SD	7 - 1
PORTCOMPRESSIONALLOWED	4 - 17
PORTCOUNTS	9 - 20
PORTLEVELINCARNATION	4 - 3
PORTLOGINTERVAL	4 - 11
PORTRESUMEREADY	4 - 18
PORTRETRYLIMIT	4 - 18
PORTSEGMENTTIMEOUT	4 - 18
PORTWINDOWSIZE	4 - 18
PROFILE	9 - 12
PROGRAM	3 - 9
PROGRAMAGENTSECURITY	3 - 8
READY HOST	6 - 10
READY STATION	6 - 8
RELEASE	1 - 16
REPORTS	1 - 16
ROUTERHEADERSIZE	4 - 3
ROUTINGREFRESH	6 - 12
ROUTINGS	9 - 8
SAVE HOST	6 - 11
SAVE STATION	6 - 9
SENDTEST	7 - 1
STARTTRACE	6 - 13
STATION	9 - 13
STOP	8 - 3
TO	6 - 14
TRANSITCOUNTLIMIT	4 - 4
TUNINGS	4 - 25
VALIDATE	3 - 6
VALIDATEANDATTACH	7 - 1
VERSION	4 - 4

APPENDIX E

COMMAND INDEX MNEMONICS, ALPHABETICAL ORDER

AC BY	6 - 1
ADD CONN.	5 -26
ADD ENS	5 -22
ADD HOST.	5 - 3
ADD NODE.	5 - 4
ADD PROF.	5 -16
ADD STA	5 -18
AT.	1 - 6
AUTH.	3 - 4
CAND.	9 -18
CC.	6 - 3
CLEAR HOST	6 -10
CLOSE CPD or SD	7 - 1
CONT.	8 - 4
DED	1 -11
DEL CONN.	5 -29
DEL ENS	5 -24
DEL HOST.	5 - 5
DEL NODE.	5 - 6
DEL PROF.	5 -17
DEL STA	5 -20
DET	7 - 1
DISCARD	8 - 5
EC.	6 - 4
ENDINIT	2 - 4
ENS	9 -17
HID	4 -21
HITO.	4 -21
HN.	1 - 5
HOST.	9 - 1
HUTO.	4 -22
LC.	6 -12
LOAD.	8 - 1
LOC	4 - 1
LOG	4 - 7
LR.	4 -16
MAXPROG	3 - 8
MHC	4 -12
MOD CONN.	5 -30
MOD ENS	5 -25
MOD STA	5 -21
MON	4 - 9
MRF	4 -12

NBTO	4 - 24
NEI	9 - 4
NET	2 - 1
NET	9 - 21
NGTO	4 - 24
NODE	9 - 4
NODESUM	4 - 26
NPW	3 - 1
NRF	4 - 13
NRTO	4 - 13
NTO	4 - 14
NW	1 - 10
NWMSS	4 - 5
NWVER	4 - 5
OPEN CPD or SD	7 - 1
PCA	4 - 17
PLI	4 - 11
PLINC	4 - 3
PORTC	9 - 20
PRL	4 - 18
PROF	9 - 12
PROG	3 - 9
PROGSEC	3 - 8
PRR	4 - 18
PSTO	4 - 18
PWS	4 - 18
READY HOST	6 - 10
READY STA	6 - 8
REL	1 - 16
REP	1 - 16
RHS	4 - 3
ROUT	9 - 8
RREFR	6 - 12
SAVE HOST	6 - 11
SAVE STA	6 - 9
SENDTEST	7 - 1
STA	2 - 13
STARTTRA	6 - 13
STOP	8 - 3
TCL	4 - 4
TO	6 - 14
TUNINGS	4 - 25
VAL	3 - 6
VALATT	7 - 1
VER	4 - 4

APPENDIX F

ERROR MESSAGE INDEX, ALPHABETICAL ORDER

ADDED STATION	5 - 25
ALL REQUIRED ATTRIBUTES NOT SET	5 - 19
ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET	2 - 4
ALL REQUIRED ATTRIBUTES HAVE NOT BEEN SET	1 - 15
ALL STATIONS IN ENSEMBLE ARE CLOSED	6 - 4
ATTRIBUTE INCONSISTENT WITH STATION TYPE.	5 - 19
AUTHORITY CANNOT BE GRANTED OR REMOVED.	3 - 6
BDLC FAILURE ON <station>	7 - 3
CALL DATA NOT REQUIRED.	6 - 7
CALL DATA REQUIRED.	6 - 7
COMMAND OUT OF SEQUENCE	7 - 3
CONNECTION IN USE	5 - 29
CONNECTION IN USE	5 - 30
DELETED STATION NOT IN ENSEMBLE	5 - 25
DUPLICATE AGENT NAME.	3 - 11
ENSEMBLE ALREADY ADDED.	5 - 23
ENSEMBLE IN USE	5 - 24
ERROR	7 - 3
FILE ALREADY SUSPENDED.	8 - 3
FILE LOADING IN PROCESS	8 - 2
HOST ALREADY ADDED.	5 - 4
HOST NAME - NODE ADDRESS DO NOT MATCH	5 - 7
HOST NAME - NODE ADDRESS DO NOT MATCH	5 - 5
HOST NAME - NODE ADDRESS DO NOT MATCH	5 - 3
HOST NAME DOES NOT MATCH.	4 - 2
HOST NOT DELETED.	5 - 5
ILLEGAL PASSWORD.	3 - 3
IMPROPER ENSEMBLE KIND.	5 - 28
IMPROPER ENSEMBLE TYPE.	6 - 2
IMPROPER ENSEMBLE TYPE.	6 - 7
IMPROPER FILE FORMAT.	8 - 2
IMPROPER PARAMETER.	5 - 30
IMPROPER PHASE.	1 - 15
IMPROPER SETTING OF STATION ATTRIBUTE	5 - 19
IMPROPER STATION KIND	5 - 28
IMPROPER STATION TYPE	6 - 2
IMPROPER STATION TYPE	6 - 7
INCORRECT PARAMETERS.	5 - 28
INVALID AUTHORIZATION OPTION.	3 - 6
INVALID HOST INACTIVE DISCONNECT OPTION	4 - 23

INVALID HOST INACTIVE TIMEOUT	4 -23
INVALID HOST NAME	5 - 7
INVALID HOST NAME	5 - 5
INVALID HOST UNREACHABLE TIMEOUT.	4 -23
INVALID LINK RESISTANCE FACTOR.	4 -17
INVALID LOGGING OPTION.	4 - 8
INVALID MAX HOP COUNT	4 -15
INVALID MAX PROGRAM AGENT VALUE	3 - 9
INVALID MAX RESISTANCE FACTOR	4 -15
INVALID MONITOR OPTION.	4 -11
INVALID MONITOR TIME INTERVAL	4 -11
INVALID NEIGHBOR BUSY TIMEOUT VALUE	4 -25
INVALID NEIGHBOR GREETING TIMEOUT VALUE	4 -25
INVALID NEIGHBOR RESTART TIMEOUT VALUE.	4 -15
INVALID NETWORK MAX SEGMENT SIZE.	4 - 6
INVALID NODE ADDRESS.	4 - 2
INVALID NODE ADDRESS.	5 - 3
INVALID NODE ADDRESS.	5 - 6
INVALID NODE ADDRESS.	5 - 4
INVALID NODE ADDRESS.	5 - 7
INVALID NODE ADDRESS.	1 -15
INVALID NODE RESISTANCE FACTOR.	4 -15
INVALID NODE UP TIMEOUT VALUE	4 -15
INVALID PORT COMPRESSION ALLOWED OPTION	4 -20
INVALID PORT LOG TIME INTERVAL.	4 -12
INVALID PORT RESUME READY FACTOR.	4 -20
INVALID PORT RETRY LIMIT.	4 -20
INVALID PORT SEGMENT TIMEOUT INTERVAL	4 -20
INVALID PORT WINDOW SIZE.	4 -20
INVALID PROFILE REFERENCE	5 -19
INVALID PROGRAM AGENT SECURITY OPTION	3 - 9
INVALID REMOTE HOST	3 -11
INVALID SLM PHASE	7 - 3
INVALID STATION ATTRIBUTE NAME.	5 -16
INVALID STATION ATTRIBUTE VALUE	5 -16
INVALID STATION NAME.	6 - 9
INVALID STATION NAME.	6 - 8
INVALID STATION PHASE ON <station>	7 - 3
INVALID STATION	5 -20
INVALID TRACE DESTINATION	6 -13
INVALID TRACE SOURCE.	6 -13
INVALID USERCODE.	3 - 6
LAST REMAINING STATION MAY NOT BE DELETED	5 -25
MESSAGE TEXT EXCEEDS MAX SEGMENT SIZE	6 -14
MISSING NODE ADDRESS.	5 - 4
NEIGHBOR IS NOT A NODE.	5 -28
NO CALLS WITH NEIGHBOR EXIST.	6 - 4
NO FILE BEING LOADED.	8 - 3
NO MORE THAN TWO PROFILES MAY BE SPECIFIED.	5 -19
NO STATION AVAILABLE.	6 - 7
NO STATION AVAILABLE.	6 - 2
NO SUCH <entity>	1 -15

NO SUCH CONNECTION.	5 - 30
NO SUCH CONNECTION.	5 - 29
NO SUCH ENSEMBLE.	5 - 25
NO SUCH ENSEMBLE.	5 - 24
NO SUCH ENSEMBLE.	6 - 2
NO SUCH ENSEMBLE.	6 - 7
NO SUCH ENSEMBLE.	6 - 4
NO SUCH HOST OR NODE.	6 - 14
NO SUCH HOST OR NODE.	3 - 3
NO SUCH HOST.	3 - 11
NO SUCH REMOTE HOST	6 - 11
NO SUCH REMOTE HOST	6 - 11
NO SUCH REMOTE HOST	6 - 10
NO SUCH STATION GROUP	6 - 2
NO SUCH STATION	6 - 2
NO SUCH STATION	6 - 4
NO SUCH STATION	6 - 7
NO SUCH USERCODE.	3 - 11
NO SUSPENDED FILE	8 - 4
NO SUSPENDED FILE	8 - 5
NODE ALREADY ADDED.	5 - 4
NODE ALREADY DELETED.	5 - 7
NODE NOT DELETED.	5 - 7
NODE NOT DELETED.	5 - 7
NODE UNREACHABLE.	6 - 14
NOT DELETED	5 - 20
NOT DELETED	5 - 20
NOT DELETED	5 - 20
NOT IN NETWORK MODE	1 - 14
<file-title> NOT PRESENT FOR LOADING.	8 - 2
PROFILE ALREADY DEFINED	5 - 16
PROFILE NOT DEFINED	5 - 17
ROUTER VALIDATE IS FALSE.	5 - 4
SHUTDOWN CANNOT BE CANCELLED.	2 - 3
STATION ALREADY CLOSED.	6 - 4
STATION ATTRIBUTE MAY NOT BE SET.	5 - 21
STATION FAILURE	6 - 3
STATION FAILURE	6 - 7
STATION NOT AVAILABLE	6 - 7
STATION NOT AVAILABLE	6 - 2
STATION NOT CLOSED.	5 - 21
STATIONS IN ENSEMBLE NOT OF PROPER TYPE	5 - 23
STATIONS IN ENSEMBLE HAVE NOT BEEN ADDED.	5 - 23
UNAUTHORIZED AGENT.	1 - 14
USERCODE MAY NOT BE SPECIFIED	3 - 11
VALIDATE OPTION INVALID	3 - 7
VALIDATION FAILURE ON <station>	7 - 3
VALIDATION FAILURE.	6 - 7
VALIDATION FAILURE.	6 - 2

APPENDIX G

REPORT INDEX, ALPHABETICAL ORDER

<neighbor> DISCONNECTED, NEIGHBOR BUSY TIMEOUT EXCEEDED . . .	10 - 6
ERROR IN MESSAGE	10 - 2
ERROR IN MESSAGE FROM	10 - 10
FROM	10 - 2
HOST SAVED.	10 - 9
HOST VALIDATION FAILURE ON	10 - 9
LOCAL VALIDATION TO <neighbor> FAILED ON <station>	10 - 7
NODE IS NOW	10 - 1
PLM SUBPORT DIALOG ESTABLISHMENT FAILURE.	10 - 9
REMOTE VALIDATION TO <neighbor> FAILED ON <station>	10 - 7
<neighbor> RESTARTING	10 - 6
ROUTING STATUS TO	10 - 5
ROUTING TRACE FROM	10 - 4
STATION	10 - 7
STATION	10 - 8
STATION	10 - 8
TEST	10 - 8

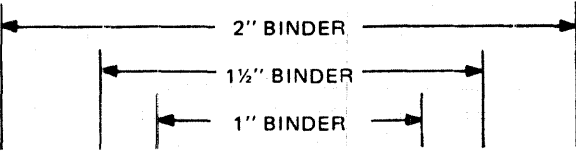
APPENDIX H

MINIMUM REQUIREMENTS FOR NETWORK OPERATION

With the exception of any system dependent requirements, the only commands necessary to invoke Network Mode are the the Host Name (HN), and a NET + command; an optional Initialization File containing at least the local node address (LOCALIDENTITY), or LOCALIDENTITY command must be provided, and the command ENDINIT must complete either procedure.

If the above minimum procedure was adopted, the node would enter the Network Mode, but would be unable to communicate with a neighbor node as there would be no stations available. Another requirement then, is to enable a station. This can be done most simply by ADDING a station, and then causing the station attachment by the command ESTABLISH CALL. Having done this, the node will be able to communicate with a neighbor, and the network.

The procedure described above is a minimum requirement, there would be no validation provided and all attributes would be set to the default values. Here it should be noted that MAX HOP COUNT and MAX RESISTANCE FACTOR default values only represent a small network. In particular MAX HOP COUNT's default value is 5. However, if the most simple case is to be used, it is most likely that these values would suffice.



Burroughs Network Architecture (BNA)
NETWORK CONTROL
REFERENCE MANUAL VOLUME 2

1132180

Printed in U.S.A.