

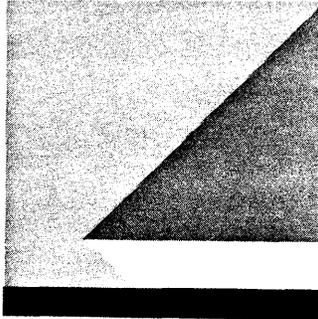


NetView™

SC31-6014-0

Administration Reference

Release 3



File Number
S370/4300/30XX-50

Program Numbers
5665-362 (MVS/XA)
5664-204 (VM)

Fifth Edition (May 1989)

This edition applies to Release 3 of the NetView™ licensed program, which runs under the following operating systems:

MVS/XA (NetView program number 5665-362)
MVS/ESA (NetView program number 5665-362)
VM (NetView program number 5664-204).

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the Agreement for IBM Licensed Programs. Changes are made periodically to the information herein; before you use this document in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program may be used.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not of itself constitute or imply a grant of any license or immunity under any patents, patent applications, trademarks, copyrights, or other similar rights of IBM or of any third party, or any right to refer to IBM in any advertising or other promotional or marketing activities. IBM assumes no responsibility for any infringement of patents or other rights of third parties that may result from use of the subject matter disclosed in this document or for the manufacture, use, lease, or sale of machines or programs described herein, outside of any responsibilities assumed via the Agreement for Purchase of IBM Machines and the Agreement for IBM Licensed Programs.

Licenses under IBM's utility patents are available on reasonable and nondiscriminatory terms and conditions. IBM does not grant licenses under its appearance design patents. Inquiries relative to licensing should be directed in writing to the IBM Director of Commercial Relations, International Business Machines Corporation, Armonk, New York, 10504.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programs, or services in your country.

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to IBM Corporation, Department E15, P.O. Box 12195, Research Triangle Park, North Carolina 27709, U.S.A. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

NetView is a trademark of International Business Machines Corporation.

© Copyright International Business Machines Corporation 1986, 1989
All Rights Reserved.

Contents

Chapter 1. Definition Statement Reference	1
Chapter 1. Definition Statement Reference	3
How to Use the Definition Statement Reference	3
Syntax Conventions for Definition Statements	4
A (Alert) Statement	5
ALIASMEM Statement	7
AUTH Statement	8
C (Command List) Statement	10
CDMNSESS Statement	11
CDRMDEF Statement	12
CMDCLASS Statement	13
CMDMDL Statement	14
CMDSYN Statement	17
CNMAUTH Statement	18
CNMTARG Statement	19
COS Statement	20
CTL Statement	21
DEFFOCPT Statement	22
DOMAINS Statement	23
DSTINIT Statement	24
F (Filter) Statement	29
HARDCOPY Statement	32
HOLDMSG Statement	33
IF-THEN Automation Statement	34
INITMOD Statement	47
ISPAN Statement	52
KCLASS Statement	53
KEYCLASS Statement	55
LOGINIT Statement	56
LU Statement	57
MAPSESS Statement	58
MAXABEND Statement	60
MAXLOGON Statement	61
MAXSPAN Statement	62
Minor Node Definitions	63
MODE Statement	64
MSGCMD Statement	65
NCCFIC Statement	66
NCCFID Statement	67
O MONIT Statement	68
OPCLASS Statement	69
OPERATOR Statement	70
OPTIONS Statement	71
ORIGNET Statement	72
PARMSYN Statement	73
PCLASS Statement	74
POS Statement	77
POSPPOOL Statement	78
PROFILE Statement	79

PROFILEN Statement	80
R (Ratio) Statement	81
RATE Statement	82
REPORTS Statement	83
RRD Statement	84
SPAN Statement	85
SPANLIST Statement	86
STATOPT Statement	87
STORPOOL Statement (MVS Only)	90
T (Timer) Statement	91
TASK Statement	92
TRANSTBL Statement	93
VALCLASS Statement	94
VPDINIT Statement	95
VSAMLSR Statement	96
W (Wrap) Statement	97

Appendixes	99
Appendix A. External Log Record Formats	101
Hardware Monitor External Log Record	101
Session Monitor External Log Record	108
Appendix B. Network Asset Management Vital Product Data Descriptions and Sample Record Formats	123
Vital Product Data Descriptions	123
Format of Records Created by Network Asset Management Sample Command Lists	131
Appendix C. Modem Configuration Change Form	137
Appendix D. Command Keywords and Values that can be Scope Checked	139
Command Facility	139
Hardware Monitor	142
Session Monitor	142
Status Monitor	143

Glossary, Bibliography, and Index	145
Glossary	147
Bibliography	169
NetView Publications	169
Other Network Program Products Publications	170
Index	173

Tables

1.	The NetView Library	x
2.	Combinations of LOG and SESSTATS	50
3.	External Log Record Header Format for Hardware Monitor	101
4.	Hardware Monitor Header Format	101
5.	Product Report Format	102
6.	Alert Report Format	102
7.	Generic Event Report Format	103
8.	Event Report Format	103
9.	Statistical Report Format	104
10.	LPDA-1 Modem Report Format	104
11.	LPDA-2 Report Format	105
12.	Local and Remote Modem Reports	105
13.	Local and Remote DSU/CSU Reports	107
14.	Local Area Network Report	107
15.	External Log Record Header Format for Session Monitor	111
16.	Data Descriptor Section for Response Time and Accounting Data Functions	112
17.	Data Descriptor Section for Storage and Event Counter Data	113
18.	Product Section	114
19.	Session Configuration Data Section	114
20.	Route Data Section	116
21.	Ten-Byte Route Element Entry	116
22.	Response Time Data Section	116
23.	Accounting and Availability Data Section	117
24.	Event Counters Data Section	118
25.	Session Awareness Counters Data Section	119
26.	Resource Counter Data Section	120
27.	Storage Counter Data Section	120
28.	Format of Common Record Prefix	132
29.	Format of Start Subrecords	132
30.	Format of End Subrecords	133
31.	Format of PU Hardware Subrecords	134
32.	Format of PU Software Subrecords	134
33.	Format of DCE Hardware Subrecords	135
34.	Format of Timeout Subrecords	136
35.	Format of Error Subrecords	136

About This Book

NetView Administration Reference describes the NetView™ program definition statements required for system administration. System administration is the process of redefining system defaults and storage requirements to improve network efficiency. This book also contains appendixes that contain external log record formats, network asset management information, a modem configuration change form, and lists of command keywords and values that can be scope checked.

This book is intended to provide the customer with an understanding of the NetView program definition statements required for system administration. It primarily contains descriptions of all the definition statements. Unless specifically stated otherwise, the information in this book must not be used for programming purposes. However, this book also provides the following types of information, which are explicitly identified where they occur: General-use programming interfaces are provided to allow the customer to write programs that use the services of the NetView program.

Who Should Use This Book

NetView Administration Reference is a reference source for system programmers and network operators who need a complete understanding of the NetView program definition statements, or need to use the information in the appendixes.

How to Use This Book

This section includes information about the organization, terms, and coding conventions used in this book.

How This Book Is Organized

This book is organized into the following sections:

Chapter 1, "Definition Statement Reference" is an alphabetical listing of all definition statements. Each entry contains an explanation of the purpose of the statement and identifies the member where it is used. The format of each entry provides all operands, default values, and information that must be supplied in the statement. See "How to Use the Definition Statement Reference" on page 3 for instructions on how to find information in this chapter.

Appendix A, "External Log Record Formats" provides external log record formats for the hardware monitor and session monitor.

Appendix B, "Network Asset Management Vital Product Data Descriptions and Sample Record Formats" provides information about the sample Network Asset Management command lists.

Appendix C, "Modem Configuration Change Form" provides a one-page form that system programmers can use to give modem configuration information to operators.

Appendix D, "Command Keywords and Values that can be Scope Checked" provides a list of the NetView commands having keywords that can be scope checked.

Notes on Terms Used in This Book

Following is a list of terms and the meanings they have in this book. Unless otherwise noted, abbreviations for products refer to the latest version and release of the product.

Term	Meaning
command lists	command lists written in REXX and command lists written in the NetView command list language
member	member in the specified data set (MVS) and file name with a file type of NCCFLST (VM)
MVS	MVS/XA, MVS/ESA (compatibility mode)
REXX	Restructured Extended Executor language
VM	VM/SP, VM/SP HPO, and VM/XA (compatibility mode)
VTAM	VTAM V3R1.1, VTAM V3R1.2, and VTAM V3R2.

Coding Conventions Used in This Book

The model statements are formatted according to a set of coding conventions which are described in this section.

Brackets []

When brackets enclose an operand, they indicate a completely optional specification. This includes any accompanying commas or equal signs. Do not include brackets when coding the specification.

OR-sign |

The OR-sign separates choices for an optional or required specification. If a group of options is enclosed by brackets and the individual options are separated by OR-signs, none of the options in the group has to be chosen. Do not include the OR-signs when coding the specification.

UPPERCASE Characters

You must spell command names or operands shown in UPPERCASE **BOLD** characters exactly as they appear. These names are program keywords.

lowercase Characters

Lowercase, *italic* characters describe the type of program variable information that must be supplied, rather than an actual value. Replace the lowercase, italic word with the actual value.

underscored Characters

Underscored characters indicate **default values**. These values are automatically assigned unless you specify a different value.

Parentheses ()

The parentheses enclose parameters that must be entered along with the parentheses.

Ellipsis ...

An ellipsis replaces a repetition of an operand or variable in syntax statements. Replace ellipses with the appropriate operand or variable when you code.

What Is New In This Book

This book describes the following additions and changes to the NetView definition statements for NetView Release 3:

- The following definition statements are new:
 - DEFFOCPT
 - RATE
 - VPDINIT.
- The following definition statements have new keywords:
 - CDRMDEF
 - CNMTARG
 - DSTINIT
 - IF-THEN
 - INITMOD.
- The RACF keyword on the OPTIONS statement has been changed to MAXIMUM.

Appendix B, "Network Asset Management Vital Product Data Descriptions and Sample Record Formats" on page 123 has been added to this book. This appendix contains information about the data returned from the VPDCMD command and the sample network asset management command lists.

Where To Find More Information

Table 1 shows all of the publications in the NetView Release 3 library, arranged according to related tasks. For more information on these and other related publications, see "Bibliography" on page 169.

Table 1. The NetView Library

Evaluation and Education

<i>Network Program Products General Information</i>	GC30-3350
<i>Bibliography and Master Index for NetView, NCP, and VTAM</i>	GC31-6081
<i>Learning about NetView: Operator Training (PC Diskettes)</i>	SK2T-0292

Planning

<i>Network Program Products Planning</i>	SC30-3351
<i>NetView Storage Estimates (PC Diskettes)</i>	SK2T-1988
<i>Console Automation Using NetView: Planning</i>	SC31-6058

Installation and Administration

<i>NetView Installation and Administration Guide</i>	SC31-6018
<i>NetView Administration Reference</i>	SC31-6014
<i>Network Program Products Samples</i>	SC30-3352
<i>NetView Tuning Guide</i>	SC31-6079

Customization

<i>NetView Customization Guide</i>	SC31-6016
<i>NetView Customization: Writing Command Lists</i>	SC31-6015
<i>NetView Customization: Using PL/I and C</i>	SC31-6037
<i>NetView Customization: Using Assembler</i>	SC31-6078

Operation

<i>NetView Operation Primer</i>	SC31-6020
<i>NetView Operation</i>	SC31-6019
<i>NetView Command Summary</i>	SX75-0026

Diagnosis

<i>NetView Problem Determination and Diagnosis</i>	LY43-0001
<i>NetView Resource Alerts Reference</i>	SC31-6024
<i>NetView Problem Determination Supplement for Management Services Major Vectors 0001 and 0025</i>	LD21-0023

Chapter 1. Definition Statement Reference

Chapter 1. Definition Statement Reference	3
How to Use the Definition Statement Reference	3
Syntax Conventions for Definition Statements	4
A (Alert) Statement	5
ALIASMEM Statement	7
AUTH Statement	8
C (Command List) Statement	10
CDMNSESS Statement	11
CDRMDEF Statement	12
CMDCLASS Statement	13
CMDMDL Statement	14
CMDSYN Statement	17
CNMAUTH Statement	18
CNMTARG Statement	19
COS Statement	20
CTL Statement	21
DEFFOCP Statement	22
DOMAINS Statement	23
DSTINIT Statement	24
F (Filter) Statement	29
HARDCOPY Statement	32
HOLDMSG Statement	33
IF-THEN Automation Statement	34
IF Keyword	35
Comparison-Expressions	35
THEN Keyword	42
Actions	42
INITMOD Statement	47
ISPAN Statement	52
KCLASS Statement	53
KEYCLASS Statement	55
LOGINIT Statement	56
LU Statement	57
MAPSESS Statement	58
MAXABEND Statement	60
MAXLOGON Statement	61
MAXSPAN Statement	62
Minor Node Definitions	63
MODE Statement	64
MSGCMD Statement	65
NCCFIC Statement	66
NCCFID Statement	67
O MONIT Statement	68
OPCLASS Statement	69
OPERATOR Statement	70
OPTIONS Statement	71
ORIGNET Statement	72
PARMSYN Statement	73
PCLASS Statement	74
POS Statement	77
POSPPOOL Statement	78
PROFILE Statement	79

PROFILEN Statement	80
R (Ratio) Statement	81
RATE Statement	82
REPORTS Statement	83
RRD Statement	84
SPAN Statement	85
SPANLIST Statement	86
STATOPT Statement	87
Using STATOPT to Define Major Nodes	89
STORPOOL Statement (MVS Only)	90
T (Timer) Statement	91
TASK Statement	92
TRANSTBL Statement	93
VALCLASS Statement	94
VPDINIT Statement	95
VSAMLSR Statement	96
W (Wrap) Statement	97

Chapter 1. Definition Statement Reference

This chapter contains a description of the definition statements used by the NetView program.

NetView definition statements are used for performing system administration tasks. System administration is the process of redefining system defaults and storage requirements. This process involves several subtasks. Administration subtasks can be performed during NetView installation, or when you redefine network resources.

In planning for installation and network management tasks, you determined the facilities you need to run NetView. You identified hardware requirements and specific resources used by NetView. This led to decisions on requirements for user coding and NetView definitions. You have performed these installation tasks:

- Completed the planning task
- Installed and verified the sample network shipped with NetView
- Determined that NetView definition is required.

Use the definition statements to accomplish the procedures described in the *NetView Installation and Administration Guide* and *Network Program Products Samples*. Those publications introduced you to the concept of altering the sample NetView program definition statements to suit your requirements. After copying some of the current definition statements from the sample files, you can begin to modify existing definition statements or create new ones. You can alter definition statements during a first-time NetView installation, or later while running your production system.

How to Use the Definition Statement Reference

You can use the following methods to locate any specific definition statement information:

1. The definition statements are listed alphabetically by statement name. When you know a statement name, look it up in alphabetical order. Use the dictionary-style headings for quick reference.
2. When you know only the member name, and want to know optional statement names not used in the samples, locate the member name in the index and find the statements of interest listed alphabetically.
3. When you know an operand name but not the statement it is used with, locate the operand name in the index.

The format of a definition statement is as follows:

- Statement name
- General introduction

The general introduction explains overall options, assumptions, and the purpose of the statement. Each introduction tells you the name of the member and data set where you code the statement.

- **Definition statement syntax**

The definition statement syntax is a model statement that is formatted according to the coding conventions described in "Coding Conventions Used in This Book" on page viii.

- **Definition of each operand**

A description of each operand that can be specified for the definition statement is provided. The specific values or variable information that can be specified for the operand are described.

- **Cross-reference to related statements**

A listing of other statements that can affect the definition statement is provided.

Syntax Conventions for Definition Statements

The following syntax conventions apply to most statements:

- You must code at least one blank between a label name and the name of the definition statement and between the name of the definition statement and the first operand. One or more blanks, or a single comma with no blanks, must separate the statement operands. The operands cannot be separated by a combination of commas and blanks.

- Continuation from one line to the next is not allowed, except for the IF-THEN statement. However, you can repeat the definition statement and add the remaining information. For example:

```
LOGINIT AUTOFLIP=YES  
LOGINIT RESUME=YES
```

is the same as:

```
LOGINIT AUTOFLIP=YES,RESUME=YES
```

- It is recommended that comments be placed on a separate line for DSIPARM members. The first column of a comment line must contain an asterisk (*). Do not place comments between the beginning and end of a continued IF-THEN statement.
- The label field can only be 8 characters long.
- Each definition file should conclude with an END statement. This END statement has no operands and must not begin in column 1.
- Command names, command list names, and any other NetView identifiers must not contain commas (,), periods (.), blanks, apostrophes ('), ampersands (&), asterisks (*), or equal signs (=). Commas, periods, blanks, and equal signs are used as delimiters when the definition statements are parsed. The other characters have special meanings for NetView command lists.

A (Alert) Statement

The A statement defines the color, highlighting, and alarm attributes corresponding to the message indicators. Messages in the network log are also displayed in the specified colors while browsing the network log data. If an alarm, color, or highlighting attribute is defined for a message, it occurs for each operator who is defined to receive the message. This statement is coded in the DSICNM member of DSIPARM.

Note: Message indicator definitions for a given operator are not enabled until the operator has been in status monitor.

Code one A statement for each message indicator class you define. You can code four message indicator class statements (1-4). If you code A statements, use IF-THEN automation statements to assign messages to the message indicator classes. You can use IF-THEN automation statements to route these messages to any NetView operators. You can also use F statements to assign messages to the message indicator classes and, in this case, the messages are always routed to the authorized message receiver.

A class color notify high alarm console comments

A must be in column 1.

class is the message indicator class number and must be from 1 through 4. The message indicator value must be in column 2 of the statement. This number is displayed at the top of any status monitor panel and is associated with a number on an F statement.

color is the color value for the message indicator class. This value must be in column 4 of the statement. The number at the top of any status monitor panel (when alerted) and the message in the network log are displayed in the chosen color. You can use the following *color* values:

B	Blue
G	Green
P	Pink
R	Red
T	Turquoise
W	White
Y	Yellow
(blank)	Default color for the terminal.

notify specifies that the *alarm* and *high* definitions are triggered for the operators when messages in this class arrive. This value must be in column 5 of the statement. You can use the following *notify* values:

Y	The <i>high</i> and <i>alarm</i> values are interpreted.
N	The <i>high</i> and <i>alarm</i> values must be blank. This is the default.

Note: If *notify* is N, then *high* and *alarm* must be blank.

high

is the highlighting value for the message indicator class. This value must be in column 6 of the statement. The number at the top of each operator's status monitor panel is highlighted according to the highlighting value.

You can use the following *high* values:

- B** Blink
- R** Reverse video
- U** Underscored
- (blank)** No highlighting. This is the default.

Note: This value must be blank if *notify* is N.

alarm

is the alarm value for the message indicator class. This value must be in column 7 of the statement. When the alarm is set, any message from the class triggers an alarm for each operator when the message arrives. The alarm is sounded only when the operator is in status monitor. You can use the following *alarm* values:

- Y** The alarm is sounded when a message arrives.
- N** The alarm is not sounded. This is the default.

Note: This value must be blank if *notify* is N.

console

indicates whether a copy of the message should be sent to the system console. This value must be in column 8 of the statement. When this value is set, any message from the class is sent to the system console. You can use the following *console* values:

- Y** A copy of the message is sent to the system console.
- N** A copy of the message is not sent to the system console. This is the default.

comments

is unused space for the statement. Any comments you want included with the statement must be in columns 9 through 80.

Related Statements: F, AUTH, IF-THEN

ALIASMEM Statement

NetView uses the ALIASMEM statement to create a table of alias name translations. This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is ALIASAPL. The precoded member used in SAMPLIB is DSIALATD.

Note: This statement must be coded after the DSTINIT statement that defines DSIAINIT as the DST initialization exit.

```
[label ]      ALIASMEM      membername [...]
```

membername [...]

names the members that contain the ORIGNET, LU, COS, and MODE definition statements.

Related Statements: COS, DSTINIT, LU, MODE, ORIGNET, TASK

AUTH Statement

The AUTH statement defines an operator's authority to control resources. This statement also specifies whether an operator can receive unsolicited messages. This statement is coded in a member of DSIPRF specified by a PROFILEN statement associated with the operator. Refer to "PROFILEN Statement" on page 80 and "OPERATOR Statement" on page 70 to see how a PROFILEN statement is associated with an operator. A precoded member used in the SAMPLIB is DSIPROFA.

```
[label ]      AUTH      [ CTL = GLOBAL|SPECIFIC ]
                  [ ,MSGRECVR = YES|NO ]
```

CTL = GLOBAL|SPECIFIC

defines the operator's authority to control resources. The value of this parameter must be one of the following:

GLOBAL means span of control is not used. DOMAINS, ISPAN, and SPAN statements are ignored. An operator with global authority can establish cross-domain sessions with domains specified by RRD statements only.

SPECIFIC means the operator can only control resources listed on ISPAN and SPAN statements and can only establish cross-domain sessions with the NCCFIDS listed on the DOMAINS statement. SPECIFIC is the default value for the CTL operand.

Note: Code MAXIMUM or NORMAL on the OPTIONS statement; otherwise, the span of control has no meaning. Refer to "OPTIONS Statement" on page 71 for more information about the OPTIONS statement.

MSGRECVR = YES|NO

specifies whether operators using this profile can receive unsolicited messages that are not routed to a particular operator by the use of the NetView ASSIGN command or by message automation.

YES means that an operator using this profile can be the authorized message receiver.

NO means that operators using the profile containing this statement do not receive unsolicited messages. NO is the default.

In NetView, the **authorized receiver** is the operator who is authorized to receive all the unsolicited and authorized messages that are not routed to a specific operator with an ASSIGN command or a ROUTE action in a message automation statement. The authorized receiver is determined by the order in which operator terminals are defined to NetView and the order in which authorized operators have logged on.

When several operators are defined to receive a particular message, NetView routes the message to the operator who meets the first of the following criteria when the message arrives:

1. The operator designated by an ASSIGN command
2. The operator or operators designated by the ROUTE action in the IF-THEN message automation statement.
3. An operator logged on to a POS terminal
If more than one POS terminal is defined, the one defined first has priority.
4. A POSPOOL operator
If more than one POSPOOL operator is logged on, the one logged on first has priority.
5. A cross-domain operator
If more than one cross-domain operator is logged on, the one logged on first has priority.
6. An autotask operator
If more than one autotask has been started, the one started first has priority. You should use the ASSIGN command if an autotask is going to be the receiver of unsolicited messages.
7. The system console operator.

Related Statements: DOMAINS, ISPAN, OPERATOR, OPTIONS, POS, POSPOOL, PROFILE, PROFILEN, SPAN

C (Command List) Statement

The `c` statement lets you assign up to 16 command lists to be displayed on the status monitor detail panel. Command lists can be written either in REXX or the NetView command list language. This statement is coded in the `DSICNM` member of `DSIPARM`.

Code one `c` statement for each command list you define. You can code up to 16 `c` statements.

```
C cmdlistname comments
```

C
must be in column 1.

cmdlistname
is the name of the command list you want displayed on the status monitor detail panel. This value must begin in column 3 of the statement.

comments
is unused space for the statement. Any comments you want included with the statement must be in columns 11 through 80.

Related Statements: None

CDMNSESS Statement

The CDMNSESS statement is used to specify the maximum number of cross-domain sessions that can be started with this NetView from other domains. If you do not code CDMNSESS for this NetView, no cross-domain sessions can be started from other domains. This statement is coded in the DSIDMN member of DSIPARM.

```
[label ]      CDMNSESS      n|0
```

n|0

is a decimal number from 0 through 32767 that specifies the maximum number of concurrent cross-domain sessions that can be started from other domains with this NetView. A logical maximum value is the number of operators in other domains who have profiles that specify this NetView in a DOMAINS statement, plus the number of global operators in other domains. The default is 0.

Note: Do not code a larger number than you need, because NetView sets aside storage for each potential session.

Related Statements: DOMAINS

CDRMDEF Statement

The CDRMDEF statement sets up CDRMNAME-NCCFID relationships for other domains. This allows the DST initialization exit for the access method LU task (DSIAMLUT), an IBM-supplied initialization exit routine, to build a table of these relationships. The table is used by the session monitor for session data retrieval from other domains. This statement can also be used to eliminate session monitor conversation set-up attempts by using the RETRY keyword. This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is DSIAMLUT. The precoded member used in SAMPLIB is DSIAMLTD.

Note: This statement must be coded after the DSTINIT statement that defines DSILINIT as the DST initialization exit.

```
[label ]      CDRMDEF      cdrmname = domainid [...]  
[RETRY = YES|NO]
```

cdrmname = domainid[...]

cdrmname specifies the 1- to 8-character name of a cross-domain SSCP known in this domain as defined to VTAM.

domainid is the 1- to 5-character NetView identifier that names the NetView application in the domain defined by *cdrmname*.

RETRY = YES|NO

defines whether an attempt to allocate a conversation will be made every 10 minutes if the first attempt fails. The default value is defined on the RETRY parameter of the DSTINIT statement. The default on DSTINIT is RETRY=YES.

YES means the session monitor attempts to establish an initial conversation with the other NetView once every 10 minutes.

NO means the session monitor attempts to establish an initial conversation with the other NetView only one time.

Note: If the CDRMDEF statement is used, it must follow the DSTINIT statement.

Related Statements: CNMTARG, DSTINIT, NCCFID, TASK

CMDCLASS Statement

The CMDCLASS statement is used to restrict the usage of a command or command list to operators defined in one or more scope classes. Refer to Appendix D, "Command Keywords and Values that can be Scope Checked" on page 139 for a list of the NetView command keywords and values that can be scope checked.

Code CMDCLASS after the CMDMDL statement of the command or command list to which it applies and before the next CMDMDL statement. Only operators without any OPCLASS statements or the specified scope class can run the specified command. If you do not code the CMDCLASS statement, the command is not limited and any operator can use it. This statement is coded in the DSICMD member of DSIPARM.

```
[label ]      CMDCLASS      scopeclass [...]
```

scopeclass [...]

specifies a scope class that can use this command. The value specified for *scopeclass* must be a number from 1 to 2040.

Note: When you define scope classes, start with 1 and work up. Using smaller numbers saves storage.

Related Statements: CMDMDL, OPCLASS

CMDMDL Statement

The CMDMDL statement defines commands and command lists to NetView. For commands, CMDMDL shows the command name, the module that processes the command, the command type, and when the command module is loaded. You need to write CMDMDL statements for command lists when:

- You want them scope checked.
- You want synonyms for them.

If a command or command list has a CMDMDL statement, then it can be scope checked using the CMDCLASS statement. Refer to Appendix D, "Command Keywords and Values that can be Scope Checked" on page 139 for a list of the NetView command keywords and their values that can be scope checked.

The CMDMDL statement is used in the DSICMD member of DSIPARM.

```

commandname          CMDMDL          MOD = modulename
                        [ ,TYPE = R||B|D|RD ]
                        [ ,CTL = N|S ]
                        [ ,RES = Y|N ]
                        [ ,ECHO = Y|N ]
                        [ ,PARSE = Y|N ]
    
```

commandname

is the name of the command or command list.

MOD = modulename

shows the name of the module that processes the command. NetView command modules begin with the letters AAU, BNJ, CNM, or DSI. For command lists, use MOD = DSICCP.

Note: When you define a user-written command procedure, be sure to specify a unique module name on the MOD operand. Do not use a name that might be recognized by the system as a command, because NetView will attempt to execute that command instead of the user-written command procedure.

TYPE = R||B|D|RD

shows the command type. If CTL = S is used to define PA or PF keys, then TYPE = I is required.

Note: Do not define regular command procedures with TYPE = I or TYPE = B.

R is a regular command. If TYPE is not coded in the CMDMDL statement, TYPE = R is assumed. Only one regular command runs at a time for an operator. If an operator enters a regular command while another regular command is running, the second command is held by NetView and runs after the first command is finished. Regular commands are allowed in command lists. This is the default.

- I** is an immediate command. An immediate command interrupts a regular command. Immediate commands are usually screen control or error recovery commands. When an immediate command is running, the keyboard is locked and other commands cannot be entered. Immediate commands are not allowed in command lists or as initial commands. They also cannot be used with the PPT operand of the AT, EVERY, and AFTER commands.
- B** is a command that can run as either a regular or an immediate command. If an operator enters the command at a terminal, NetView treats it as an immediate command. If the command is in a command list, NetView treats it as a regular command.
- D** is a data services command. Data services command procedures run under a data services task (DST). Because these commands are internal, operators cannot enter them at their terminals. Some internal commands that support the session and hardware monitors are TYPE=D.
- RD** is a regular or data services command. An example is the SWITCH command.

CTL = N|S

controls whether the command is normal (N) or special (S). If CTL=S is used to define PA or PF keys, then TYPE=I is required.

N means that the command is normal. N is the default.

S means that the command is special. Special means that PA or PF keys are defined as commands. For commands you want to be able to set to a key, you must use MOD=DSIPFK.

Note: Do not use CMDMDL statements to define PF keys to regular command procedures.

RES = Y|N

specifies whether the command module is loaded when NetView is initialized. At initialization, when NetView loads only frequently used command modules, you save storage. However, if you load all command modules at initialization, performance can be degraded because less operating storage is available.

Y means the command module is loaded at NetView initialization and remains resident in storage. Y is the default.

N means the command module is not loaded until the command is run, and the storage that is used is freed upon completion.

If the following commands are coded RES=N, NetView automatically makes them RES=Y:

- All VTAM commands
- DSICCP (command lists command processor)
- AGAIN, BGNSSESS, DSIYCLF, DSIYCTRM, DSIYOCRM, DSIYOFPM, DSIYOLRP, DSIYOMRP, DSIYOSRP, DSIYOTP, DSIYPIF, LOGOFF, REPLY, START, STOP.

ECHO = Y|NO

controls whether a command is echoed to the screen after an operator enters it. For commands in command lists, the setting for &CONTROL determines the echo status. TYPE=I and TYPE=B commands are not echoed after they are entered from a terminal.

Y means an entered command is echoed. Y is the default.

N means an entered command is not echoed.

Notes:

1. Code ECHO=N for all commands and command lists that invoke full-screen panel displays.
2. Commands that are echoed to the screen are subject to message automation if they match the conditions in an IF-THEN automation statement.

PARSE = YN

specifies whether to parse the message buffer.

Note: This operand is recognized only by DST processing.

Y means to parse the message buffer. Y is the default.

N means that the message buffer is not parsed. PARSE=N improves command processor performance for user-written command procedures that do not use the parse buffer PDB. For information on how to write command procedures, see *NetView Customization: Using Assembler*.

Related Statements: CMDCLASS, CMDSYN, KEYCLASS, PARMSYN, VALCLASS

CMDSYN Statement

The CMDSYN statement allows you to define additional names for commands or command lists. This statement is coded in the DSICMD member of DSIPARM.

Multiple CMDSYN and PARMSYN statements can follow each CMDMDL statement. The CMDSYN statements must follow all of the PARMSYN statements for a CMDMDL; otherwise, the parameter synonyms for that CMDMDL are ignored. CMDSYN statements must also precede any CMDCLASS, KEYCLASS, or VALCLASS statements for the command.

Notes:

1. Do not use the following as command synonyms:
 - A comma or a period
 - A name that begins with a percent sign
 - A name that is a VTAM command
 - Another NetView command
 - A command in an application program that runs with NetView.
2. NetView help is keyed to command names. Therefore, if you create a synonym, you must create help for that synonym. Refer to *NetView Customization Guide* for more information on writing help panels.
3. The synonyms might not work as expected depending on the NetView component you are using. For example, you can assign the T synonym to the TARA command using the CMDSYN statement; however, if you are inside the hardware monitor, you will execute the TITLE command instead of the TARA command.

```
[label ]      CMDSYN      synonym
```

synonym

is the new name for the CMDMDL statement that precedes this CMDSYN statement. The operator can use either the original command name or the new synonym. However, the system console operator must always use the original command name.

Related Statements: CMDCLASS, CMDMDL, KEYCLASS, PARMSYN, VALCLASS

CNMAUTH Statement

The CNMAUTH statement defines the authorization requirements for the CNM data transfer sessions. This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is xxxxxLUC (xxxxx is the NetView identifier specified by the DOMAINID keyword of the NCCFID statement). The precoded member used in SAMPLIB is DSILUCTD.

Note: This statement must be coded after the DSTINIT statement that defines DSILINIT as the DST initialization exit.

```
[label]          CNMAUTH          [ CTL = GLOBAL|SPECIFIC ]
                                   [ ,MAXSESS = n ]
```

CTL = GLOBAL|SPECIFIC

defines authorization requirements for LU sessions.

GLOBAL means there is no authority check when NetView establishes new cross-domain session monitor sessions. The system accepts any session initiation request. GLOBAL is the default.

SPECIFIC indicates that sessions are initiated and accepted only for specific LUS. Specific LUS are those that are defined on the CNMTARG statement.

MAXSESS = n

specifies the number of parallel sessions the NetView CNM data transfer task is authorized to establish with another NetView LU. The maximum value you can specify for MAXSESS is 65535. If you do not code the MAXSESS operand, NetView does not set a limit to the number of sessions in which it is the primary logical unit.

Related Statements: CNMTARG, NCCFID, TASK

CNMTARG Statement

The CNMTARG statement defines authorized LU names for the CNM data transfer task. If you specify GLOBAL on the CNMAUTH statement, NetView ignores the specific CNMTARG LU names specified in the CNMTARG statements. If you specify SPECIFIC on the CNMAUTH statement, you need a CNMTARG statement for each remote NetView or NCCF that communicates with this NetView. You define an LU name only if you need to start a session with that LU.

You can use the PERSIST keyword to specify whether a session between this NetView and a remote NetView or NCCF is persistent. A persistent session is one that remains active regardless of the time elapsed between conversations between this NetView and a remote NetView or NCCF. A non-persistent session is one that is terminated if it is inactive for longer than the value specified in the non-persistent sessions timeout interval in DSICTMOD. The default value of the timeout interval is zero. See *NetView Installation and Administration Guide* for information on changing the value of the timeout interval.

This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is xxxxxLUC. (xxxxx is the NetView identifier specified by the DOMAINID keyword of the NCCFID statement.) The precoded member used in SAMPLIB is DSILUCTD.

Note: This statement must be coded after the DSTINIT statement that defines DSILINIT as the DST initialization exit.

```
[label ]      CNMTARG      LU = luname
                [,PERSIST = YES|NO]
```

LU = luname

is the 4- to 8-character APPL name for the remote CNM data transfer task. CNM data transfer tasks are named xxxxxLUC, where xxxxx is the NetView identifier of the NetView where the CNM data transfer task runs. You can have multiple CNMTARG statements. To view session data from another NetView, you must be authorized by that NetView.

PERSIST = YES|NO

defines whether a session between this NetView and a remote NetView or NCCF is persistent. The default value is defined on the PERSIST parameter of the DSTINIT statement. The default on DSTINIT is PERSIST=YES.

YES means that the session is persistent and remains active regardless of the time elapsed between conversations between this NetView and a remote NetView or NCCF.

NO means that the session is non-persistent and will terminate if the time elapsed between conversations between this NetView and a remote NetView or NCCF reaches CTM_MSGACT_INTVL seconds.

Related Statements: CDRMDEF CNMAUTH, DOMAINS, LU, NCCFID, TASK

COS Statement

The COS statement defines equivalent names for duplicate class of service (COS) names in an interconnected network. This statement is coded in the members of DSIPARM specified in the ALIASMEM statement. The precoded member used in SAMPLIB is DSIALTAB.

```
[label]      COS      targname,targnet,origname
```

targname

is the COS name that applies to the target network. This name is also called the equivalent name because it becomes equivalent to the COS name in the origin network.

targnet

is the 1- to 8-character name of the network in which the COS is known by its equivalent name.

origname

is the name the COS is known by in the origin network.

Related Statements: ALIASMEM, ORIGNET

CTL Statement

The CTL statement specifies the number of wrap count statements (*www*) and ratio statements (*rrr*) that follow. This statement is closely associated with two other statements: *w* and *R*. The *w* statement defines the initial wrap values. The *R* statement defines the ratio values.

This group of statements contains input data for BNJAPAMA and establishes initial wrap counts and user alert thresholds. This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is BNJDSESV. The precoded member used in SAMPLIB is BNJMBDST.

Note: This statement must be coded after the DSTINIT statement that defines BNJAPAMA as the initialization exit.

```
[label ]      CTL      www rrr
```

www

defines the number of wrap count statements you need to code after the CTL statement.

Note: This value must be a 3-digit number. Use leading zeros if necessary.

rrr

defines the number of ratio statements you need to code after the CTL statement.

Note: This value must be a 3-digit number. Use leading zeros if necessary.

Related Statements: DSTINIT, R, TASK, W

DEFFOCPT Statement

The DEFFOCPT statement defines the focal point nodes for a given NetView hardware monitor. Focal point alert support makes centralized problem determination easier by providing centralized alert notification. If more than one DEFFOCPT statement is coded, the last one is used.

Note: This statement is valid only in the DSICRTD initialization member.

```
[label]      DEFFOCPT      PRIMARY=nodename
                [BACKUP=nodename2]
```

PRIMARY = *nodename*

the eight-character name of the LUC task of the primary focal point. LUC tasks are named xxxxxLUC, where xxxxx is the NetView identifier of the NetView in which the LUC task runs.

BACKUP = *nodename2*

the eight-character name of the LUC task of an optional backup focal point. LUC tasks are named xxxxxLUC, where xxxxx is the NetView identifier of the NetView in which the LUC task runs.

Note: If a primary focal point is not active and a backup focal point is active, the backup focal point will become the primary focal point.

Related Statements: None

DOMAINS Statement

The DOMAINS statement allows the setup of cross-domain communication for operators with specific control. This statement is coded in the operator's profile. It shows which cross-domain sessions this operator can start. This statement is coded in the member specified by the PROFILEN statement associated with the operator. Refer to "OPERATOR Statement" on page 70 and "PROFILEN Statement" on page 80 to determine how a PROFILEN statement is associated with an operator.

```
[label ]          DOMAINS          domainid [...]
```

domainid [...]

is a 1- to 5-character identifier of a NetView or NCCF in another domain, where this operator can start a cross-domain session. The NetView identifiers are coded on the NCCFID definition statements for the other domains.

The DOMAINS statement can be used as many times as necessary to define all the required domains. The DOMAINS statement is ignored if either AUTH CTL=GLOBAL or OPTIONS VERIFY=MINIMAL is specified.

Related Statements: AUTH, CDMNSESS, NCCFID, OPERATOR, OPTIONS, PROFILEN

DSTINIT Statement

The DSTINIT statement defines the initialization parameters required by optional data services tasks (DST). This statement is coded in the members of DSIPARM specified by the MEM keyword on the TASK statements where MOD=DSIZDST. The pre-coded members used in SAMPLIB are: AAUCNMTD, AAUPRMLP, BNJMBDST, BNJ36DST, DSIALATD, DSIALMLTD, DSICPINT, DSICRTD, DSIELMEM, DSILOGBK, DSILUCTD, DSITRCBK, and USRSQMEM.

```
[label]          DSTINIT          [FUNCT = VSAM|CNMI|BOTH|OTHER ]
                                     [,PBSDN = name ]
                                     [,PDDNM = name ]
                                     [,UNSOL = name ]
                                     [,PPASS = password ]
                                     [,SBSDN = name ]
                                     [,SDDNM = name ]
                                     [,SPASS = password ]
                                     [,DSRBU = nnn|5 ]
                                     [,DSRBO = nnn|3 ]
                                     [,XITBN = modname [...]]
                                     [,XITBO = modname [...]]
                                     [,XITDI = modname [...]]
                                     [,XITCO = modname [...]]
                                     [,XITCI = modname [...]]
                                     [,XITVN = modname [...]]
                                     [,XITVI = modname [...]]
                                     [,XITVO = modname [...]]
                                     [,XITXL = modname [...]]
                                     [,MACRF = NSR|LSR|DFR ]
                                     [,PERSIST=YES|NO]
                                     [,RETRY=YES|NO]
```

FUNCT = VSAM|CNMI|BOTH|OTHER

specifies the task function. Do not code multiple DSTINIT statements with conflicting FUNCT definitions.

VSAM shows that the task is to use VSAM only. UNSOL is ignored if VSAM is specified. DSRBU is set to 0 if VSAM is specified.

CNMI specifies that this task will send and receive CNM data to and from the network only. UNSOL need not be specified unless unsolicited CNM data is to be processed by the DST. A primary data set need not be specified.

BOTH specifies that the task is to send and receive CNM data to and from the network and use VSAM.

OTHER specifies that this task does not use VSAM or CNM services. For example, this operand is used for the VTAM LU task or to define a sequential log task. DSRBU is ignored, if it is specified, and is set to 0.

PBSDN = name

is the 1- to 8-character DD or FILEDEF name of the primary data set to be used by NetView sequential log services. For VM, the name is 1- to 7- characters. You must enter this name on a DD or FILEDEF statement when you start NetView or

anytime before starting the sequential log tasks through the NetView ALLOCATE command or the VM FILEDEF command. This keyword should only be specified once in each sequence of DSTINIT statements if you use NetView sequential log services. The PBSDN keyword is required for sequential logging and causes NetView to define the task as a sequential task.

PDDNM = name

is the 1- to 8-character DD name of the primary data set to be used by VSAM services. For VM the name is 1- to 7-characters. You must enter this name on a DD statement when you start NetView. Specify PDDNM at least once in each sequence of DSTINIT statements if you use VSAM.

UNSOL = name

is the 1- to 8-character verb that is the label of the CMDMDL statement for the data services command processor (DSCP) that gains control when unsolicited CNM data is received from the network, or passed to the DST by the CNM router subtask (DSICRTR).

PPASS = password

is a 1- to 8-character VSAM password to be used when the primary data set ACB is opened. This password must match the UPDPW parameter in the primary VSAM data base allocation.

SBSDN = name

is the 1- to 8-character DD or FILEDEF name of the secondary data set to be used by NetView sequential log services. For VM, the name is 1- to 7-characters. You must enter this name on a DD or FILEDEF statement when you start NetView or anytime before starting the sequential log tasks through the NetView ALLOCATE command or the VM FILEDEF command. This keyword should only be specified once in each sequence of DSTINIT statements if you use NetView sequential log services. The SBSDN keyword is not required for sequential logging.

SDDNM = name

is the 1- to 8-character DD name of the secondary data set to be used by VSAM services. For VM, the name is 1- to 7- characters. You should define this name on a DD statement when you start NetView.

SPASS = password

is the 1- to 8-character password to be used when the secondary data set ACB is opened. This password must match the UPDPW operand in the secondary VSAM data base allocation.

DSRBU = nnn|5

is a decimal number from 0 through 999 that specifies the number of data services request blocks (DSRBS) to be pre-allocated for processing unsolicited problem determination request units (RUS). The default value is 5.

Note: If the task is an LUC, AM/LU, DSITRACE, or DSILog, NetView sets DSRBU to 0.

DSRBO = nnn|3

is a decimal number from 1 through 999 that specifies the projected number of concurrent user requests for services from this DST. The value represents the number of DSRBS to be pre-allocated for processing solicited RUS and VSAM requests. If more requests are received than there are DSRBS available, the requests are queued. The default value is 3.

Notes:

1. If the task is an LUC, AM/LU, DSITRACE, or DSILOG, NetView sets DSRBO equal to 1.
2. If the task is DSICRTR (router), the default in the CNMS0057 sample is 5. However, this value may need to be increased as indicated in the CNMS0057 sample.

The following operands that begin with XIT are 1- to 8-character load module names for user-written exit routines for this DST. You can code up to 10 module names for each exit. These modules receive control in the order they are coded in the DST.

XITBN = modname[,...]

receives control when a sequential log data set is opened for output processing only (not for a resume). This exit allows you to put an initialization record into the file.

XITBO = modname[,...]

receives control before each NetView sequential log services record is blocked for output. XITBO can only use the service facilities available to a DST sub-task defined as FUNCT=OTHER, excluding the DSIWLS macro.

XITDI = modname[,...]

receives control as each DST initialization parameter is read. This exit routine processes statements following this statement only. DSTINIT XITDI must precede any statement that is to be processed by the DST initialization exit.

XITCO = modname[,...]

receives control for CNM interface output processing before the request is passed to VTAM.

XITCI = modname[,...]

receives control for CNM interface input processing before data received is passed to the DSCP. Specify this parameter under the DSICRTR subtask for access to unsolicited CNM data prior to NetView internal routing.

XITVN = modname[,...]

receives control when an empty VSAM data set is opened for processing. This exit allows you to put an initialization record into the file. If XITVN is not specified, access method services (AMS) must be used to start the data set.

XITVI = modname[,...]

receives control on input from the VSAM file before the input record is passed to the DSCP.

XITVO = modname[,...]

receives control before output of a record to the VSAM file.

XITXL = modname[,...]

receives control before data is logged to an external log. If SMF (MVS only) is not used, this exit can perform the necessary logging. XITXL receives control by coding the DSIWLS macro with the EXTLOG keyword. For VM, see DSIELXIT in the NetView SAMPLIB for an example of a user exit that logs data to a file.

MACRF = NSR|LSR|DFR

specifies local resource sharing. Specify the option that meets the requirements of your installation.

NSR is the level of support provided in NCCF. This is the default.

LSR enables the reclaiming of data and index buffers by keeping a pool of the most recently referenced records in storage. This is effective in reducing physical I/O.

DFR extends LSR to defer writing of records. DFR defers the writing of a record until NetView forces it out, because buffer space is needed for a read. This further reduces I/O by minimizing writes.

Note: If MACRF=DFR is defined, do not use the MVS operator RESET command to stop NetView because records that are deferred might not be logged, which corrupts the data base. Use the NetView CLOSE or CLOSE IMMED command instead.

You can use DFR and LSR with the VSAM Local Shared Resource (LSR) option. LSR is explained in the *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*. The possible values indicate the following:

DFR = Buffer pool scanning and deferred writes

LSR = Buffer pool scanning on input only.

If you are not using LSR, do not code this statement. See the *NetView Installation and Administration Guide* for more information about using the VSAM LSR option with NetView.

Note: NetView supports the LSR or DFR facility for the network log as well as the standard VSAM support for CNM applications. Do not use LSR or DFR for the DSILOG task if you wish to browse the network log.

PERSIST = YES|NO

defines the default for whether all sessions between this NetView and a remote NetView or NCCF are persistent. For more information on persistent sessions, see "CNMTARG Statement" on page 19. Code this keyword in the member identified by the MEM keyword of the TASK statement that has a TASKID of XXXXXLUC.

YES means that the sessions are persistent and remain active regardless of the time elapsed between conversations between this NetView and a remote NetView or NCCF.

NO means that the sessions are non-persistent and terminate if the time elapsed between conversations between this NetView and a remote NetView or NCCF reaches the non-persistent sessions timeout interval in DSICTMOD.

Note: The PERSIST parameter of the CNMTARG statement can override this default for specific LUS.

RETRY = YES|NO

defines the default for whether an attempt to allocate a conversation between this NetView and all the domains specified by CDRMDEF statements will be made every 10 minutes if the first attempt fails. Code this keyword in the member identified by the MEM keyword of the TASK statement that has a TASKID of DSIAMLUT.

YES means the session monitor attempts to establish an initial conversation with the other domains once every 10 minutes.

DSTINIT

NO means the session monitor attempts to establish an initial conversation with the other domains only once.

Note: The RETRY parameter of the CDRMDEF statement can override this default for specific domains.

Related Statements: ALIASMEM, CDRMDEF, CNMAUTH, CNMTARG, CTL, INITMOD, LOGINIT, R, REPORTS, TASK, W

F (Filter) Statement

The F statement lets you associate messages to the message indicator classes you defined with the A statement. Based on the tests you specify, a message indicator can be triggered at the authorized receiver. The NETLOG action of the IF-THEN automation statement can also be used to trigger message indicators. See "IF-THEN Automation Statement" on page 34 for more information. The F statement is used in the DSICNM member of DSIPARM.

Notes:

1. Messages DSI531I, DSI559I, DSI546I, and DSI547I are routed to all NetView operators who are logged on. If an operator is in a terminal access facility (TAF) session when a message is sent, the message is not sent to the operator's screen until the TAF session is exited. At that time, the filter and alert in the status monitor is set off. Depending on how much time has elapsed while the operator was in the TAF session, the circumstances under which the message was issued may have changed and the message may no longer be relevant.
2. When the VTAM function MSGMOD=ON is coded, VTAM inserts a five-character module name into all IST messages between the message number and the message text. If MSGMOD is coded as ON, the F statements you code for IST messages will have unpredictable results.

Code as many F statements for each message as you like. You can code F statements only for message indicator classes 1 through 4.

```

F  class      msgnum      &e operator 'string '
F                                     continuation expressions

                                     or

F  class      msgnum      modname &e operator 'string '
F                                     continuation expressions

```

F is the statement name and must be in column 1.

class

is the message indicator class number and must be from 1 through 4. This value must be in column 5 of the statement. This number corresponds to the class number on the A statement.

msgnum

is the first parsed element of a message (&1). This value must begin in column 7 of the statement.

Note: You can code as many F statements for a message number as you require. NetView classifies the message based upon the first true F statement for the message number. A single F statement can contain several comparison expressions. Comparison expressions are separated by commas. All of the comparisons on a single F statement must be true for the classification to be made.

&e

the element selected for comparison to *string*. The expression must be located in columns 16 through 72. If the expression must continue onto successive lines, make a comma the last character on a line.

&D Domain identifier

&O Operator identifier

&T Time (hh:mm:ss)

&M The HDRMTYPE from DSITIB. See *NetView Customization: Using Assembler* for more information on the message header type.

&n The number of the element as parsed by NetView. This number must be from 1 to 255. &1 is the message number.

operator

the logical symbol used to determine the relationship between **&e** and *string*. The following logical symbols are allowed:

= **&e** is equal to *string*.

≠ **&e** is not equal to *string*.

> **&e** is greater than *string*.

→> **&e** is not greater than *string*.

< **&e** is less than *string*.

→< **&e** is not less than *string*.

'*string*'

the value or string you want compared with **&e**.

continuation expression

the continuation of an expression from the preceding line. Begin the continuation statement with an F and then continue the expression in columns 16 through 72.

modname

the name of an optional module that processes **&e** and returns a value that NetView will compare to *string*. *modname* must begin in column 16.

NetView is delivered with the module CNMFANGC, which returns a one-character string that represents the type of node. The type of node depends on the value of **&e**. You can write other modules to meet your needs or use CNMFANGC or both. Your module or CNMFANGC is invoked with the following registers:

REG1 Address of a parameter list

REG13 Address of the caller's save area

REG14 Return address of the calling program

REG15 Entry-point address of *modname*.

The parameter list passed in REG1 contains five addresses:

Word	Address	Description
1	MVTADDR	The MVT address.
2	TXTPTR	The address of &e.
3	TXTLEN	The address of a one-byte field containing the length of &e.
4	RESPTR	The address of the field containing the result of <i>modname</i> processing. (For CNMFANGC, this is one of the node types described in the following table.)
5	RESLEN	The address of a one-byte field containing the length of the result (pointed to by RESPTR). This word is set to the length of <i>string</i> and cannot be altered. (CNMFANGC uses this word to blank out RESPTR before putting anything in it.)

As an example, CNMFANGC takes &e (a node name) and converts it to one of the following one-character node types:

H Host	E Local PU
N NCP major node	D Local LU
L Line	B Application major node
C PU/cluster	A Application minor node
T LU/terminal	Y CDRM major node
S Switched major node	Z CDRM
R Switched PU	W CDRSC major node
Q Switched LU	X CDRSC
F Local major node	BLANK No node found

This result is then compared to *string* to determine if an alert should be set up.

Related Statements: A, IF-THEN

HARDCOPY Statement

The HARDCOPY statement designates the devices that can be used for hard-copy printers. It is used in the DSIDMN member of DSIPARM.

```
[label ]      HARDCOPY      devicename [...]
```

devicename [...]

specifies the name of the printer as it is defined to VTAM. This name can be 1- to 8-characters long.

Use as many HARDCOPY statements as you need to define all your hard-copy printers.

Although each operator can be assigned to only one hard-copy printer, several operators can share the same one. However, if too many operators share the same hard-copy printer, messages for that printer can accumulate and might not be printed for some time after they are received.

Related Statements: None

HOLDMSG Statement

The HOLDMSG statement is not needed for NetView Release 3. This statement causes NetView to hold the message on the operator's screen. If the message you are trying to hold is part of a multi-line write-to-operator (MLWTO) message, HOLDMSG must identify the message that is the first line of the MLWTO. The entire MLWTO is held.

```
[label]      HOLDMSG      msgid [...]
```

msgid [...]

is a 5- to 8-character message identifier of a message from any product, command list, or user-written application that runs with NetView.

You should not use the HOLDMSG statement in NetView Release 3. In NetView Release 1, this statement was used in the DSIHMF member of DSIPARM. The DSIHMF member of DSIPARM is not needed in NetView Release 3. Once you have started NetView, changes to the DSIHMF member of DSIPARM will not go into effect until you start NetView again.

The HOLDMSG statement is supported in NetView Release 3 for compatibility with NetView Release 1. In NetView Release 3, the same function is provided by specifying HOLD(Y) on the IF-THEN automation statement. For each message you want held, code a statement such as:

```
IF MSGID='msgid' THEN HOLD(Y);
```

where *msgid* is the message identifier.

Note: The HOLDMSG statement holds a message for both the primary and secondary message receivers. The IF-THEN statement with just HOLD(Y) specified holds a message for the primary receiver only. To use IF-THEN to hold a message for both primary and secondary receivers, specify EXEC ROUTE(ALL *opids*) and HOLD(Y) on the IF-THEN statement, where *opids* is the list of primary and secondary message receivers. For example:

```
IF MSGID='ABCD' THEN EXEC ROUTE(ALL OPER1, OPER2, +GRP1) HOLD(Y);
```

Related Statements: IF-THEN

IF-THEN Automation Statement

The IF-THEN statement allows you to specify the messages you want NetView to intercept and process without operator action. You can use this statement to code the conditions that must be met to select a message for processing, and the actions you want NetView to take if a message meets those conditions. You must write a separate IF-THEN automation statement for each message or group of messages you want NetView to process automatically.

You can store your IF-THEN automation statements in one or more of the members you created in DSIPARM. To activate IF-THEN statement processing, you must issue an AUTOMSG command that specifies the DSIPARM message automation member you want to use. This member becomes your active NetView message automation member and is compiled by the AUTOMSG command into a message automation table. For more information about NetView message automation members, see *NetView Installation and Administration Guide*.

When NetView receives a message and message automation is active, NetView searches the active message automation table from the first statement to the last, looking for conditions that match the received message. NetView performs the THEN actions of the first statement that has IF conditions matching the message. Once NetView finds IF conditions that match a message, it stops searching the message automation table. Therefore, if you have more than one IF-THEN automation statement that could match a certain message, place the statements with more specific conditions earlier in the message automation member, followed by the statements with less specific conditions.

Syntax Notes:

1. You must code a semicolon (;) at the end of each IF-THEN automation statement.
2. You can use blanks to indent lines and to separate keywords, logical operators, and parentheses. However, blanks used within a comparison string are valid characters in that string.
3. You can continue an IF-THEN automation statement on as many lines as needed, using columns 1 through 72. You can stop a line after any logical operator, a parenthesis, a completed *comparison-expression*, or an operand, and resume the statement anywhere in the first 72 columns of the next line.
4. You must code all characters in an IF-THEN automation statement in uppercase.
5. It is recommended that comments be placed on separate lines for DSIPARM members. Do not put comment lines between the beginning and end of a continued IF-THEN automation statement. When coding a comment line, the first column must contain an asterisk (*).

The format of the IF-THEN automation statement is:

```
IF comparison-expressions THEN actions;
```

IF Keyword

You must code the IF keyword at the beginning of each IF-THEN automation statement.

Comparison-Expressions

NetView evaluates the *comparison-expressions* that you coded in the IF portion of an IF-THEN statement based on a received message. If the *comparison-expressions* are true, the *actions* coded in the THEN portion of the IF-THEN statement are executed.

If you want to take certain actions when NetView receives a multi-line write-to-operator (MLWTO) message, your *comparison-expressions* must describe the first line of the message. The *actions* you specify apply to the entire multi-line message group.

You can link multiple *comparison-expressions* together by using either a logical-and (&) or a logical-or (|) operator. When *comparison-expressions* are linked with the & operator, *all* of the linked *comparison-expressions* must be true. For example, the following IF-THEN statement has two *comparison-expressions* linked with the & operator:

```
IF DOMAINID='CNM01' & TEXT='PURGE DATE IS LATER THAN TODAY''S DATE'  
THEN EXEC (CMD('CLISTA') ROUTE (ONE * OPER1));
```

For the *comparison-expressions* to be true, the message must originate in domain CNM01, and its text must be PURGE DATE IS LATER THAN TODAY'S DATE.

When *comparison-expressions* are linked with the | operator, *any* of the linked *comparison-expressions* must be true. For example, the following IF-THEN statement has two *comparison-expressions* linked with the | operator:

```
IF DOMAINID='CNM01' | TEXT='PURGE DATE IS LATER THAN TODAY''S DATE'  
THEN EXEC (CMD('CLISTA') ROUTE (ONE * OPER1));
```

The *comparison-expressions* are true if the message originated in domain CNM01 or if its text is PURGE DATE IS LATER THAN TODAY'S DATE.

NetView evaluates expressions linked with & before those linked with |. For example, the following IF-THEN statement has three *comparison-expressions* linked with | and & operators:

```
IF DOMAINID='CNM01' | TEXT='PURGE DATE IS LATER THAN TODAY''S DATE'  
& SYSID='MVS' THEN EXEC (CMD('CLISTA') ROUTE (ONE * OPER1));
```

When processing this IF-THEN statement, NetView evaluates the TEXT and SYSID *comparison-expressions* first, because they are linked with the logical-and (&) operator, then evaluates the DOMAINID *comparison-expression*. This means both the TEXT and SYSID *comparison-expressions* must be true or the DOMAINID *comparison-expression* must be true.

IF-THEN

You can control the order in which *comparison-expressions* are evaluated by using parentheses to group comparisons that you want evaluated together. If, in the previous example, you want NetView to evaluate the DOMAINID and TEXT *comparison-expressions* first, code parentheses around them, as follows:

```
IF (DOMAINID='CNM01' | TEXT='PURGE DATE IS LATER THAN TODAY'S DATE')  
& SYSID='MVS' THEN EXEC (CMD('CLISTA') ROUTE (ONE * OPER1));
```

When processing this IF-THEN statement, NetView evaluates the DOMAINID and TEXT *comparison-expressions* first, because they are grouped within parentheses, then evaluates the SYSID *comparison-expression*. This means either the DOMAINID or the TEXT *comparison-expressions* must be true *and* the SYSID *comparison-expression* must be true.

The three kinds of comparison-expressions are:

- *Parse-function*
- *Token-function*
- *Bitstring-function*.

Parse-function Comparison-Expressions

Use the *parse-function* comparison-expression to compare alphanumeric character strings. The format is:

```
parse-name [(pos)] = | ≠ compare-item
```

parse-name

The following *parse-names* can be used. These *parse-names* can also be assigned to a variable in the IF portion of an IF-THEN automation statement. The variable can then be used in the *cmdstring* parameter of the THEN portion of an IF-THEN automation statement. See the explanation of the *cmdstring* parameter on page 42.

DOMAINID

indicates that you want to compare the NetView domain where the received message originated to *compare-item*. The *compare-item* is the 1- to 8-character name of a NetView domain.

HDRMTYPE

indicates that you want to compare the header type of the received message to *compare-item*. The *compare-item* is a 1-character message header type. See the description of the BUFHDR control block in *NetView Customization: Using Assembler* for information on the possible values of HDRMTYPE.

JOBNAME

indicates that you want to compare the name of the MVS job where the received message originated to *compare-item*. The *compare-item* is the 1- to 8-character name of an MVS job. JOBNAME can only be used when executing NetView in an MVS environment.

Because the JOBNAME is the name of the job that originated the message, it may not always be the same as the name of the job to which the message is referring. For example, this can occur when MVS issues a message about the NetView job. Also, JOBNAME can contain the name of an initiator (instead of the actual jobname)

when a job is started or terminated. If the message is issued during startup or termination, extract the job name from the message text rather than using the `JOBNAME` variable. The value of `JOBNAME` is null if the message is not received through the subsystem interface (SSI).

MSGID

indicates that you want to compare the message identifier of the received message to *compare-item*. The *compare-item* is a 1- to 10-character message identifier. The message identifier of the received message is the first ten characters of the first token of the message. If a reply ID is sent with the message, it is not used as the first token.

SESSID

indicates that you want to compare the identifier of the TAF session from which the received message was sent to *compare-item*. The *compare-item* is a 1- to 8-character TAF session identifier.

SYSID

indicates that you want to compare the identifier of the MVS system where the received message originated to *compare-item*. The *compare-item* is a 1- to 8-character MVS system identifier. `SYSID` can only be used when executing NetView in an MVS environment. The value of `SYSID` is null if the message is not received through the subsystem interface (SSI).

TEXT

indicates that you want to compare the text of the received message to *compare-item*. `TEXT` contains the entire message text. The *compare-item* is a 1- to 255-character string.

(pos)

specifies the position within *parse-name* where you want the comparison to begin. The default value is 1.

`= | \neg =`

`=` indicates that *parse-name* must equal *compare-item* for the *comparison-expression* to be true.

`\neg =` indicates that *parse-name* must not equal *compare-item* for the *comparison-expression* to be true.

compare-item

The *compare-item* for `DOMAINID`, `HDRMTYPE`, `JOBNAME`, `MSGID`, `SESSID`, and `SYSID` *parse-names* must be either a *literal* or a *variable-name*. The *compare-item* for the `TEXT` *parse-name* can be any combination of *literals*, *variable-names*, and *place-holders*.

literal

an alphanumeric character string that is compared with the message you want checked for automatic processing.

IF-THEN

Enclose a *literal compare-item* in single quotes. When using a single quote as part of the comparison string, code a second single quote following the first. For example, to compare a message such as:

```
PURGE DATE IS LATER THAN TODAY'S DATE
```

code the following:

```
'PURGE DATE IS LATER THAN TODAY''S DATE'
```

Do not substitute a double quote for two single quotes.

The maximum length allowed for a *literal compare-item* is 255 characters. Try to keep *literal compare-items* on one line. If you must use multiple lines for a single *compare-item*, do not indent the continuation lines. End each line in column 72, and begin each continuation line in column 1.

You can also continue a *literal compare-item* by breaking it into smaller *literal compare-items* on multiple lines. Consecutive *literal compare-items* are concatenated without blanks being added, so you do not have to end the lines in column 72 and begin them in column 1. For example, you could continue a *literal compare-item* on multiple lines as follows:

```
IF TEXT='PURGE DATE IS LATER '  
        'THAN TODAY''S DATE'
```

variable-name

a variable name is used to designate part of the message information that NetView should skip over during comparison processing but store for later use during *action* execution.

If you want NetView to skip over part of the message text during comparison processing, you can assign part of the text to a *variable-name*. The text that was skipped over is stored in the *variable-name* you specify and can be used in the *cmdstring* parameter of the THEN portion of the IF-THEN automation statement.

For example, the following IF-THEN statement contains the *variable-name* DATEVAR:

```
IF DOMAINID='CNM01' & TEXT='DATA BASE HASN'T BEEN PURGED SINCE'  
DATEVAR THEN EXEC (CMD('CLISTA' DATEVAR) ROUTE ALL OPERA OPERB);
```

If NetView receives the following message:

```
DATA BASE HASN'T BEEN PURGED SINCE 5/3/88
```

then this statement says to do the following:

1. Store the value of the text following the word SINCE in the DATEVAR variable
2. Run CLISTA under both OPERA and OPERB using the value of DATEVAR as a parameter:

```
CLISTA 5/3/88
```

If you want NetView to use some other message information in the *cmdstring* parameter of the THEN portion of an IF-THEN statement, you can assign the value of the following *parse-names* to a *variable-name*:

- DOMAINID
- HDRMTYPE
- JOBNAME
- MSGID
- SESSID
- SYSID.

For example, the following IF-THEN statement contains the *variable-name* DOMID:

```
IF TEXT='PURGE DATE IS LATER THAN TODAY' 'S DATE' & DOMAINID=DOMID
THEN EXEC (CMD('CLISTA 'DOMID) ROUTE ALL OPERA OPERB;
```

If NetView receives the following message from domain CNM01:

```
PURGE DATE IS LATER THAN TODAY' 'S DATE
```

then this statement says to do the following:

1. Store the value of DOMAINID in the DOMID *variable-name*. The value of variable DOMID becomes CNM01.
2. Run CLISTA under both operators OPERA and OPERB using DOMID as a parameter:

```
CLISTA CNM01
```

The maximum length of a *variable-name* is 16 characters. You can code up to 25 *variable-names* in any IF-THEN automation statement using any name not also used in the same statement as a keyword, command name, or command list name. Do not use the same variable name more than once in any one IF condition.

place-holder

a period (.) acts as a place holder and causes NetView to skip over part of the message text that does not need to be compared. The following IF-THEN statement uses *place-holders* to cause NetView to skip over parts of the message text:

```
IF TEXT = . 'SENSE CODE=' SENSE .
THEN EXEC (CMD('CLIST1 'SENSE) ROUTE ONE OPERA OPERB);
```

If NetView receives the following message:

```
RESOURCE LU1 SENSE CODE=08 NOT ACTIVATED
```

this statement says to do the following:

1. Skip over all of the text preceding SENSE CODE=
2. Store the value '08' in the *variable-name* SENSE
3. Skip over all of the text following the *variable-name* SENSE.

NetView *does not* store text designated by a *place-holder* for later use. Use a *variable-name compare-item* to store text for later use.

Notes:

1. A period, coded within single quotes, is treated as the period character, not as a *place-holder*.
2. A *place-holder* differs from a period used in an &IF control statement in the NetView command list language. In an &IF control statement, a period is used to avoid any problems that would occur if a variable has a null value.

Token-function Comparison-Expression

The *token-function* comparison-expression is used to compare words and phrases. The format is:

TOKEN (*token-number* [*pos*]) = | \neg = *compare-item*

TOKEN

instructs NetView to divide the message into tokens (words and phrases) before comparing it to *compare-item*. When comparing message text "token-by-token", NetView uses the blank space(s) between words and phrases to divide the message into tokens. All of the characters between two blank spaces equals one token.

(*token-number* [*pos*])

token-number identifies the token in the message text that is to be compared to *compare-item*. *token-number* must have a numeric value. The default value is 1.

pos identifies the position within a token of the message text where comparison with *compare-item* should begin. *pos* can be any number from 1 to 255. The default value is 1.

= | \neg =

= indicates that *token-number* must equal *compare-item* for the *comparison-expression* to be true.

\neg = indicates that *token-number* must not equal *compare-item* for the *comparison-expression* to be true.

compare-item

compare-item for a *token-function* can be a *literal* or *variable-name* and is coded the same as the *parse-function compare-item*. See the explanation of the *compare-item* parameter in the *parse-function* variable on page 37.

The following sample IF-THEN automation statement uses a *token-function* as its second comparison expression:

```
IF DOMAINID='CNM01' & TOKEN (5 3)  $\neg$  = 'OPER1' THEN DISPLAY(N);
```

This statement tells NetView that the message should not be displayed if it originated in domain CNM01 and if its fifth token does not contain the operator identifier OPER1, beginning in position three.

Bitstring-Function Comparison-Expression

The *bitstring-function* specifies one or more routing or descriptor codes in the message that you want checked against your automatic processing criteria. The format is:

bitstring-name [(*pos*)] = | \neg = *bitstring*

bitstring-name

bitstring-name can be one of the following:

ROUTCDE

identifies one or more MVS routing code bits assigned to the message that should be compared with *bitstring*. A message can have up to 128 routing code bits. The value of ROUTCDE is null if the message is not received through the subsystem interface (SSI).

DESC

identifies one or more MVS descriptor codes assigned to the message that should be compared with *bitstring*. Each message has 16 descriptor code bits. The value of DESC is null if the message is not received through the subsystem interface (SSI).

See *MVS System Programming Library: System Macros and Facilities, Vol. 2* for more information on route and descriptor codes.

(pos)

when *bitstring-name* is ROUTCDE, *pos* specifies the message routing code bit where the *bitstring* comparison should begin. The *pos* parameter can be any number from 1 to 128. The default value is 1.

when *bitstring-name* is DESC, *pos* specifies the message descriptor code bit where the *bitstring* comparison should begin. The *pos* parameter can be any number from 1 to 16. The default value is 1.

= | ¬=

= indicates that *bitstring-name* must equal *bitstring* for the *comparison-expression* to be true.

¬= indicates that *bitstring-name* must not equal *bitstring* for the *comparison-expression* to be true.

bitstring

bitstring can have any combination of the values 1, 0, and X. Enclose *bitstring* in single quotes. If *bitstring-name* is ROUTCDE, *bitstring* can be up to 128 characters long. If *bitstring-name* is DESC, *bitstring* can be up to 16 characters long.

A value of 1 in *bitstring* tells NetView to check that the routing or descriptor code bit has been assigned to the message, and a value of 0 tells NetView to check that the code bit has *not* been assigned to the message. An X value tells NetView *not to check* the value of that routing or descriptor code bit in the message. For example, if you were checking for a bit string of 0X1, the bit strings 011 and 001 would both match.

The following example IF-THEN automation statement uses ROUTCDE and DESC comparison-expressions:

```
IF ROUTCDE (3) = '10011' | DESC (6) = '110'
THEN EXEC (ROUTE (ALL OPER1 OPER4 OPER6)) BEEP(Y);
```

This statement tells NetView that when the message routing code has the numbers 10011 starting in the third position *or* when the message descriptor code has the numbers 110 in positions 6 to 8, then the message is routed to the operators whose identifiers are OPER1, OPER4, and OPER6, and an audible alarm is sounded on their terminals when the message is displayed.

IF-THEN

THEN Keyword

The THEN keyword defines the *actions* in each IF-THEN automation statement.

Actions

In the THEN portion of an IF-THEN statement, you can specify one or more *actions* for NetView to take when the IF conditions of the IF-THEN statement are true. The two kinds of actions are:

- *Execute-action*
- *Yes-or-no-action*.

Execute-Actions

An *execute-action* is one or more executable actions. You can code multiple *execute-actions* on a single IF-THEN automation statement. The format is:

```
EXEC ([CMD(cmdstring)] [ROUTE(routeparms)])
```

EXEC

tells NetView to execute the following *actions*. Either CMD or ROUTE or both can be specified with each EXEC. You can specify EXEC more than one time in a single IF-THEN statement to execute multiple commands and route multiple copies of the message.

CMD

tells NetView you want a command or command list executed.

(*cmdstring*)

contains the complete command syntax of the command or command list. The command and any *literal* parameters must be enclosed in single quotes. Any *variable-names* you want to use as parameters must be outside the single quotes. Any delimiters that need to be included in the command string also need to be enclosed in single quotes. For example, in the following *execute-action*:

```
EXEC (CMD('CLISTB ' VARPARAM1 ' ' VARPARAM2 ' LITPARM'))
```

a space is included within the single quotes after CLISTB to delimit the VARPARAM *variable-name*. A space is enclosed in single quotes between VARPARAM1 and VARPARAM2 to delimit these two *variable-names*. Also, a space is included within the single quotes before the LITPARM literal parameter. If the value of VARPARAM1 were OPER4 and the value of VARPARAM2 were OPER5, this example would execute as:

```
CLISTB OPER4 OPER5 LITPARM
```

For information on passing variables from the IF portion of an IF-THEN statement to a command or command list, see the *parse-function* section on page 36.

ROUTE

instructs NetView to route the *cmdstring* or the message to the operators whose identifiers are specified in *routeparms*. ROUTE enables you to route the *cmdstring* or message to different tasks.

Note: See *NetView Customization: Writing Command Lists* for information on the MSGROUTE command. MSGROUTE can be used when you need to check the value of global variables or the message text of a line other than the first in a

multi-line write-to-operator message before deciding where to route the message.

(*routeparms*)

specifies the operators or groups of operators to whom the message or *cmdstring* should be routed for processing. The format is:

```
ONE|ALL [*] [PPT] [oper[...]] [+grp[...]]
```

Note: At least one operand must follow either ONE or ALL.

ONE|ALL

ONE means the *cmdstring* or message is routed to the first operator in the list who is logged on to NetView or to the first logged-on operator assigned to the group if a group appears in the list.

ALL means the *cmdstring* or message is routed to all the operators and groups of operators in the list who are logged on to NetView.

*

means the *cmdstring* or message is routed to the current operator task (the task where the message is intercepted for automation checking).

PPT

means the *cmdstring* or message is routed to the PPT task for processing.

oper [...]

the identifier of the operators to whom the *cmdstring* or message should be routed. The operator identifier must be defined to NetView with an OPERATOR definition statement (see "OPERATOR Statement" on page 70). The maximum length of an operator identifier is 8 characters. You can code as many operator identifiers as needed.

+*grp* [...]

the identifier of the groups of operators to whom the *cmdstring* or message should be routed. The maximum length of a group identifier is 8 characters, and it must begin with a plus (+) sign. You can code as many group identifiers as needed. Define group identifiers with the ASSIGN command. See *NetView Operation* for more information about the ASSIGN command.

The following example IF-THEN automation statement has two *execute-actions*:

```
IF MSGID='AAU0001' THEN
EXEC (ROUTE(ALL * OPER1 OPER2)
EXEC (CMD('CLIST1') ROUTE(ONE OPER3 OPER4);
```

When message AAU0001 is received, copies of the message are sent to the current operator task, OPER1, and OPER2. CLIST1 is executed under OPER3 if OPER3 is logged on to NetView. If OPER3 is not logged on to NetView, CLIST1 executes under OPER4 if OPER4 is logged on.

Notes:

1. It is recommended that when you specify CMD on an EXEC, that you also specify ROUTE. If an EXEC specifies CMD but not ROUTE, where NetView executes the CMD action is determined as follows:
 - If message automation is invoked under an OST, the CMD action is executed under that OST.

IF-THEN

- If message automation is invoked under the PPT, the CMD action is executed under the PPT. Message automation is invoked under the PPT if one of the following types of messages are received for which no ASSIGN PRI is in effect and no authorized receiver is logged on:
 - Unsolicited VTAM messages
 - Authorized receiver messages
 - Command output of commands running under the PPT
 - Messages originating in the PPT task
 - Messages queued to the PPT task.
 - For messages that are unsolicited, unassigned, and received over the subsystem interface (SSI), the following happens:
 - If CNMCSSIR (the SSI router task) was started by a START command, the CMD action is executed under the task that started CNMCSSIR. Therefore, you should always start CNMCSSIR from an autotask and not the PPT or a regular OST.
 - If CNMCSSIR was started by the INIT=Y parameter of the TASK statement, the CMD action is not executed, and the message is discarded. Therefore, you should never use INIT=Y on the TASK statement for CNMCSSIR.
2. If an EXEC specifies ROUTE but not CMD, NetView routes the message to the operators specified in *routeparms*, using the BEEP, DISPLAY, HCYLOG, and HOLD *yes-or-no-actions* specified in the IF-THEN statement. See “Yes-Or-No-Actions” for information about *yes-or-no-actions*.
 3. If an EXEC specifies both CMD and ROUTE, NetView executes the CMD actions under the operators specified in *routeparms*. Commands cannot be routed to SYSOP or the NetView log.

NetView executes the BEEP, DISPLAY, HCYLOG, and HOLD *yes-or-no-actions* under the current operator task.

ROUTE does not provide any portion of the message to the *cmdstring* parameter. Use the *parse-names* from the IF portion of the statement to pass message information to the *cmdstring* parameter.

Yes-Or-No-Actions

The *yes-or-no-actions* are actions, which you specify, for NetView to take when the IF conditions of an IF-THEN statement are true. The format is:

```
action-name(Y|N)
```

action-name(Y|N)

You can code each *action-name* only one time for an IF-THEN statement. Once an *action-name* is specified, it cannot be changed for different *execute-actions*. An *action-name* can be one of the following:

BEEP

determines whether an audible alarm sounds when the message is displayed (see note 2). If BEEP is not specified, the default value is BEEP(N).

DISPLAY

determines whether the message is displayed (see note 2). If DISPLAY is not specified, the default value is DISPLAY(Y).

HCYLOG

determines whether the message is placed in the hardcopy log (see note 2). If HCYLOG is not specified, the default value is HCYLOG(Y).

HOLD

determines whether the message is held on the operator's screen after it is displayed (see note 2). If HOLD is not specified, the default value is HOLD(N). Some messages can already have a HOLD attribute attached to them. If you code a HOLD *action-name* for these messages, the HOLD *action-name* overrides the attached HOLD attributes. If you do not code a HOLD *action-name* for these messages, their attached HOLD attributes are used instead of the NetView HOLD=NO default.

NETLOG(N)(Y [*indicator-number*] [*] [*oper*[,...]] [+*grp*[,...]]])

determines whether the message is placed in the NetView log and whether the message activates a status monitor important message indicator for specified operators or groups of operators. If NETLOG is not specified, the default value is NETLOG(Y).

Note: The F (FILTER) statement may also be used to trigger important message indicators. See "F (Filter) Statement" on page 29 for more information.

<i>indicator-number</i>	identifies the status monitor important message indicator.
*	means the message is logged as important for the operator task that the message is routed to or the current operator task (the task where the message is intercepted for automation checking).
<i>oper</i> [...]	the operator identifier of the operators for whom the message is logged as important. The operator identifier must be defined to NetView with an OPERATOR definition statement (see "OPERATOR Statement" on page 70). The maximum length of an operator identifier is 8 characters. You can code as many operator identifiers as needed.
+ <i>grp</i> [...]	the group identifier of the groups of operators for whom the message is logged as important. The maximum length of a group identifier is 8 characters, and it must begin with a plus (+) sign. Define group identifiers with the ASSIGN command. See <i>NetView Operation</i> for more information about the ASSIGN command.

If the operator has not been in status monitor or log browse but is logged on, message CNM039I is displayed:

```
CNM039I AN IMPORTANT MESSAGE HAS BEEN LOGGED -
PLEASE BROWSE THE NETVIEW LOG.
```

If only an *indicator-number* is specified, the message is logged as important for the authorized receiver. The following example shows how the IF-THEN statement is coded with only an indicator-number:

```
IF MSGID='IST105I' THEN NETLOG(Y 2);
```

Message IST105I is defined as an important message with a status monitor important message indicator number of 2. When NetView encounters message IST105I, the message is logged as important for the authorized receiver.

If an *indicator-number* and a list of operators or groups of operators are specified, the message is logged as important for the operators and groups of operators listed. The following example shows how message IST105I is logged when an *indicator-number* and a list of operators and groups of operators are specified:

```
IF MSGID='IST105I' THEN NETLOG(Y 2 * OPER1 +GRP5 OPER6);
```

Message IST105I is defined as an important message with a status monitor important message indicator number of 2. When NetView encounters message IST105I, it is logged as important with defined highlighting for OPER1, OPER6, all of the operators assigned to group +GRP5, and the current operator. If operator OPER1 and OPER6 are also assigned to group +GRP5, each operator receives only one copy of message CNM039I (if they have not been in STATMON), and duplicate highlighting and logging do not occur.

SYSLOG

determines whether the message is placed in the MVS system log. If SYSLOG is not specified, the default value is SYSLOG(N). SYSLOG has no effect on messages received from the subsystem interface (SSI). Messages received from the SSI are unconditionally placed in the MVS system log before being sent to NetView. SYSLOG can only be used when executing NetView in an MVS environment.

For an example of the coding for *yes-or-no-actions*, see the sample IF-THEN statements shown on page 40.

Notes:

1. If ROUTE is specified but not CMD, NetView routes the message to the operators specified in *routeparms* using the BEEP, DISPLAY, HCYLOG, and HOLD *yes-or-no-actions* specified in the IF-THEN statement. Otherwise, NetView executes the BEEP, DISPLAY, HCYLOG, and HOLD *yes-or-no-actions* under the current operator task (the task where the message is intercepted for automation checking).
2. NetView processes HOLD and BEEP *actions* only when DISPLAY(Y) is coded. HOLD and BEEP do not work for messages routed to an MVS console.
3. A YES or NO setting for DISPLAY, HCYLOG, NETLOG, or SYSLOG on the OVERRIDE command overrides the settings specified for these *actions* on the IF-THEN statement. A DISABLE setting for BEEP or HOLD on the OVERRIDE or DEFAULTS command means the settings specified in the IF-THEN statement for these *actions* are not used. See *NetView Operation* for more information on the OVERRIDE and DEFAULTS commands.
4. You can code YES or Y for yes, and NO or N for no.

Related Statements: ALERT, ASSIGN, F, HOLDMSG, OPERATOR

INITMOD Statement

The INITMOD statement specifies the installation parameters to the session monitor. The statement is coded in the member of DSIPARM specified by the TASK statement whose TSKID is AAUTSKLP. The value of the MEM keyword of the TASK statement identifies the member of DSIPARM that contains the INITMOD statements. The precoded member used in SAMPLIB is AAUPRMLP.

Note: This statement must be coded after the DSTINIT statement that defines AAUNIT as the DST initialization exit in AAUPRMLP.

Note: A single INITMOD statement is also coded in the member of DSIPARM specified by the TASK statement whose TSKID is DSISVRT.

```
[label ]      INITMOD      AAUICPEX
INITMOD      AAUICPEX
INITMOD      AUTHDOM = NONE|ANY|domainid
INITMOD      AAUINLDM
INITMOD      AUTHORIZ = NONE|ANY|netid
INITMOD      AAUINLDM MAXEND = nnn|100
INITMOD      AAUINLDM BUFTYPE = SAW
INITMOD      [,BUFSIZE = bbK|4K][,BUFNUM = ccc|2]
INITMOD      AAUINLDM BUFTYPE = TRACE
INITMOD      [,BUFSIZE = bbK|4K][,BUFNUM = ccc|2]
INITMOD      AAUINLDM SAW = YES|NO
INITMOD      AAUINLDM TRACELU = YES|NO
INITMOD      AAUINLDM TRACESC = YES|NO
INITMOD      AAUINLDM RTM = YES|NO
INITMOD      AAUINLDM RTMDISP = YES|NO
INITMOD      AAUINLDM
INITMOD      LOG = YES|NO[,SESSTATS = YES|NO]
INITMOD      AAUINLDM KEEPPIU = nnn|10
INITMOD      AAUINLDM KEEPDISC = nnn|250
INITMOD      AAUINLDM KEEPRTM = nnn|10
INITMOD      AAUINLDM NETID = networkname
INITMOD      AAUINLDM PERFMEM = membername
INITMOD      AAUINLDM KEEPMEM = membername
INITMOD      AAUINLDM LUCOUNT = nnn|16000
```

AAUICPEX

must be the first INITMOD statement.

Note: This is the only INITMOD statement used for the task whose TSKID is DSISVRT.

AAUICPEX AUTHDOM = NONE|ANY|domainid

initializes cross-domain authorization. You can code as many of these statements as you require for your environment. The possible values of AUTHDOM indicate the following:

NONE indicates that nobody is authorized to establish a cross-domain session with this NetView.

ANY indicates that anyone can establish a cross-domain session with this NetView.

domainid indicates that anyone using the specified NetView can establish a cross-domain session with this NetView.

The default is ANY. Specifying NONE provides security for both the SDOMAIN and the TRACE with domain ID commands.

AAUINLDM AUTHORIZ = NONE|ANY|*netid*

specifies whether NetView operators in other networks can view session configuration and trace data collected by this NetView. As many as 255 AUTHORIZ statements can be coded. The values of AUTHORIZ have the following meaning:

NONE means that no NetView operator in another network is authorized to view session configuration and trace data collected by this NetView.

ANY means that NetView operators in all other networks are authorized to view session configuration and trace data collected by this NetView.

netid means that NetView operators in the specified network are authorized to view session configuration and trace data collected by this NetView.

The default is ANY. Specifying NONE may not provide security in all configurations.

AAUINLDM MAXEND = *nnn*|100

specifies the number of concurrent requests for PIU trace data, where *nnn* is the maximum number you can specify. The value of *nnn* may be from 1 through 999; the default is 100. When the value of *nnn* is exceeded, you will receive message AAU0811. If you receive the message frequently, increase the value of *nnn*. If the specified value is too low, an operator viewing the trace data might not see the most recent PIU data from an active session.

AAUINLDM BUFTYPE = SAW[,BUFSIZE = *bbK*|4K][,BUFNUM = *ccc*|2]

are the number and size of session awareness (SAW) data buffers. For BUFSIZE, *bbK* can have values from 2K (K equals 1024) to 32K; the default is 4K. For BUFNUM, *ccc* can have values from 2 to 255; the default is 2.

Note: Although BUFTYPE is specified with a keyword, it is a positional operand and must be specified first as shown. The other operands are keywords that can be specified in any order or omitted.

AAUINLDM BUFTYPE = TRACE[,BUFSIZE = *bbK*|4K] [,BUFNUM = *ccc*|2]

are the number and size of PIU trace data buffers. For BUFSIZE, *bbK* can have values from 2K to 32K; the default is 4K. For BUFNUM, *ccc* can have values from 2 to 255; the default is 2.

Note: Although BUFTYPE is specified with a keyword, it is a positional operand and must be specified first as shown. The other operands are keywords that can be specified in any order or omitted.

AAUINLDM SAW = YES|NO

specifies whether session awareness (SAW) data should be collected. If this statement is omitted, session awareness data is collected, beginning at NetView initialization time.

YES indicates that session awareness (SAW) data is collected, beginning at NetView initialization time. This is the default value.

NO indicates that no session awareness data is collected until the ENABLE command is entered.

AAUINLDM TRACELU = YES|NO

specifies whether NetView should start tracing LU information at NetView initialization.

YES indicates that LU-LU sessions are traced beginning at NetView initialization.

NO indicates that no LU tracing occurs until the TRACE START command is entered. This is the default value.

AAUINLDM TRACESC = YES|NO

specifies whether NetView should start tracing SSCP information at NetView initialization.

YES indicates that NetView starts tracing SSCP information at NetView initialization.

NO indicates that NetView does not start tracing SSCP information until the TRACE START command is entered. This is the default value.

AAUINLDM RTM = YES|NO

enables the NetView response time monitor function.

YES indicates that NetView response time monitor function is enabled.

NO indicates that NetView response time monitor function is not enabled. This is the default value.

AAUINLDM RTMDISP = YES|NO

specifies whether you can display your response times.

YES indicates that you can display your response times.

NO indicates that you cannot display your response times.

Notes:

1. This value overrides the value specified in the hardware configuration.
2. The DSPLYLOC parameter of the PCLASS statement overrides this value.

AAUINLDM LOG = YES|NO [,SESSTATS = YES|NO]

specifies whether NetView writes records to an external log and whether the network accounting and availability function is active. The external log can be the SMF log (MVS only). For VM, logging is made to a sequential CMS file by way of a user-written exit routine. (The NetView SAMPLIB contains a sample routine you can use or modify for this purpose.) Valid combinations are listed in Table 2.

Table 2. Combinations of LOG and SESSTATS

If you specify	NetView writes to external log
LOG = YES, SESSTATS = YES	<ul style="list-style-type: none"> • Response time data • Configuration data • Accounting and availability data (session statistics) • Session start records • Session end records for any sessions with response time data monitoring.
LOG = YES, SESSTATS = NO	<ul style="list-style-type: none"> • Response time data • Configuration data • Combined session start-end records for any sessions with response time data monitoring.
LOG = NO, SESSTATS = NO	<ul style="list-style-type: none"> • No session monitor data. This is the default.

See "DSTINIT Statement" on page 24 for additional NetView definition statements that are required for external logging. See Appendix A, "External Log Record Formats" on page 101 for the record formats NetView writes to the external log.

AAUINLDM KEEPIU = *nnn*|10

is the number of PIU trace data records to be kept in virtual storage for all sessions. This number can be from 1 through 999; the default is 10. The KEEPIU value can be overridden by defining keep classes. You can also change the KEEPIU value for an individual session by using the KEEP PIU command. For more information on KEEP, see *NetView Operation*.

AAUINLDM KEEPDISC = *nnn*|250

is the number of discarded PIU trace data records to be kept in virtual storage. This number can be from 1 through 999; the default is 250.

AAUINLDM KEEPRTM = *nnn*|10

is the number of response time collection periods that can be kept in virtual storage for a specific session. This number can be from 1 through 999; the default is 10. RTM keep counts can be changed only at initialization.

AAUINLDM NETID = *networkname*

is the 1- to 8-character name of this network. This name is required if you have an interconnected network and have not defined a NETID to VTAM. This name is also required if you are using VTAM V2R1 in an interconnected network.

See *Network Program Products Planning* for more information about interconnected networks and NETID.

You must be consistent in the manner in which you define NETID in your network. Each NetView or NLDM in a network must have the same NETID, whether it is defined to NetView, to NLDM, or to VTAM.

If your network is not an interconnected network, you can omit this statement from all NetViews and NLDMS in your network.

AAUINLDM PERFMEM = *membername*

is the 1- to 8-character member name for performance class definitions for response time monitor. If performance classes are defined, this statement specifies the member name that contains the PCLASS and MAPSESS definition statements. The member is in the DSIPARM data set.

Omit this statement if you do *not* use NetView to collect RTM data.

AAUINLDM KEEPMEM = *membername*

is the 1- to 8-character member name for keep class definitions. If keep classes are defined, this statement specifies the member name that contains the KCLASS and MAPSESS definition statements. The member is in the DSIPARM data set.

Omit this statement if you are *not* using keep classes.

AAUINLDM LUCOUNT = *nnn*16000

is the number of logical units (LUS) in this network. This is a performance tuning value, and need not be exact. The number can be from 1 through 999999. The default is 16000.

When you assign a value, be sure to include all LUS owned by the local system services control point (SSCP) as well as all LUS owned by other SSCPs that can have session partners owned by the local SSCP.

Related Statements: DSTINIT, KCLASS, MAPSESS, PCLASS

ISpan Statement

The ISpan statement sets up the initial span of control for the operator. The operator can remove this span of control by entering the NetView STOP command with the SPAN operand. This statement is coded in the member of DSIPRF specified by the PROFILEN statement associated with the operator. Refer to "OPERATOR Statement" on page 70 and "PROFILEN Statement" on page 80 to determine how a PROFILEN statement is associated with an operator.

```
[label]      ISpan      spanname [...]
```

spanname [...]

is the 1- to 8-character name of the span. You can code as many span names and as many ISpan statements as you need to specify all of your spans of control. The span names coded on ISpan are associated with span names coded on the SPANLIST statement in NetView and the SPAN operand in VTAM.

Related Statements: OPERATOR, PROFILE, PROFILEN, SPAN, SPANLIST

KCLASS Statement

The KCLASS statement defines keep classes. A keep class determines what data is kept for the sessions mapped into that keep class. A session is mapped into a keep class when the session starts (session awareness). Once a session is mapped into a keep class, it remains there until the session ends.

Code one KCLASS statement for each keep class you define. You can code as many KCLASS statements as you want.

If you code KCLASS and MAPSESS statements, create a member in the DSIPARM data set to put them in. The name for this new member must match the name coded on the INITMOD AAUINLDM KEEPMEM = *membername* statement. The KCLASS statements must be the first statements in the member.

The KCLASS and MAPSESS statements control the amount of session awareness data recorded. If you do not code any of these statements, data for all sessions is recorded according to the defaults for the KCLASS statement.

```
cname      KCLASS      [ SAW = YES|NO ]
                [ ,KEEPPIU = nnn|10 ]
                [ ,DASD = NO|YES( [ ,FAILURES ] [ ,DATA ]
                [ ,BINDFAIL ] [ ,INITFAIL ] [ ,RTMDATA ]
                [ ,SESSFAIL ] [ ,SESSNORM ] [ ,TRACDATA ] )
```

cname

is the name of the defined keep class. This name is used on one or more MAPSESS statements as the value of *cname* in the KCLASS parameter. The name must be from 1- to 8-characters long and must start in column one of the statement. This is a required parameter.

SAW = YES|NO

defines whether session awareness data should be kept for the sessions mapped into this keep class.

YES keeps session awareness data.

NO does not keep session awareness data.

Session awareness data is always kept for SSCP-SSCP sessions and SSCP-PU sessions. To ensure that session awareness data is kept for all sessions, including SSCP-LU and LU-LU, specify SAW=YES. To collect response time monitor data for LUS attached to terminal control units equipped with the response time monitor feature, you must keep session awareness data for both SSCP-LU and LU-LU sessions.

Note: If you specify SAW=NO, INIT failure and BIND failure data is still kept if the INITFAIL and BINDFAIL parameters are coded under the DASD keyword of the KCLASS statement. However, if SAW is coded as NO in the INITMOD statement, no session awareness data is kept. This applies to all session types as well as INIT failures and BIND failures.

Keep session awareness data for all cross-network sessions. This ensures that cross-network data retrieval functions work correctly.

KEEPPIU = *nnn*|10

determines the initial PIU keep count for the sessions mapped into this keep class. The value of *nnn* must be from 1 through 999. The default is 10. Once a session is active, the number of PIUs kept for that session can be changed dynamically using the KEEP PIUS operator command.

DASD = NO|YES

defines the conditions under which the sessions mapped into this keep class are recorded to the VSAM file.

NO does not record sessions to the VSAM file.

YES records the following specified sessions to the VSAM file.

- FAILURES** records LU-LU sessions to the VSAM file only when they end with an abnormal UNBIND reason code or when they fail to BIND.
- DATA** records sessions to the VSAM file only if there is some trace or RTM data to record with the session.
- BINDFAIL** records sessions to the VSAM file if there is a failure to BIND (BIND failure).
- INITFAIL** records sessions to the VSAM file if there is a failure to set up prior to BIND flow (INIT failure).
- RTMDATA** records sessions to the VSAM file if there is some RTM associated with the session.
- SESSFAL** records sessions to the VSAM file if there is an abnormal UNBIND reason code.
- SESSNORM** records sessions to the VSAM file if there is a normal UNBIND reason code.
- TRACDATA** records sessions to the VSAM file if there is some trace data associated with the session.

Note: Sessions are always recorded when a FORCE command is issued by an operator, regardless of how DASD is specified. See *NetView Operation* for additional information about the FORCE command.

Related Statements: INITMOD, MAPSESS, PCLASS

KEYCLASS Statement

The KEYCLASS statement associates a command keyword with one or more scope classes. Operators who have these scope classes coded in an OPCLASS statement in their profiles or who have global control (no OPCLASS statements coded) can run a NetView command that uses this keyword. Code KEYCLASS after the CMDMDL statement of the command to which it applies and before the next CMDMDL statement. This statement is coded in the DSICMD member of DSIPARM.

```
keyword[=OTHER] KEYCLASS [scopeclass[,...]]
```

keyword[=**OTHER**]

keyword specifies a keyword that is scope-checked for the command shown on the previous CMDMDL statement. If you renamed a command operand using PARMSYN, you must use the original name on the KEYCLASS statement.

See Appendix D, "Command Keywords and Values that can be Scope Checked" on page 139 for a list of keywords and keyword values that can be scope checked.

= OTHER lets you check the scope for all other keywords that have not specifically been entered for a command.

Note: You can code multiple KEYCLASS statements, but =OTHER can be coded only once and only on the last KEYCLASS statement.

scopeclass [...]

specifies a scope class that can use the keywords. Scope class must be a number from 1 through 2040. If no scope class is specified, any operator can use this keyword.

Note: If you want to code the scope of commands for a keyword value, you must code the KEYCLASS statement and the VALCLASS statement. If you want the scope of commands for the keyword value and not for the keyword itself, code KEYCLASS without any *scopeclass*.

Related Statements: CMDMDL, OPCLASS, PARMSYN, VALCLASS

LOGINIT Statement

The LOGINIT statement defines the initialization parameters for the network log, trace logs, and optional NetView sequential log(s). The AUTOFLIP parameter specifies whether NetView should automatically continue recording on the primary data set when the secondary data set becomes full. The RESUME parameter specifies whether to begin recording at the beginning of the primary data set or to start where the previous session left off. For the most efficient logging, define both AUTOFLIP and RESUME to YES.

Note: If you allocate only one data set, logging stops when it is full. However, if you define a secondary data set, it is used automatically.

This statement is coded in the members of DSIPARM specified by the MEM keyword of the TASK statement whose TASKIDS are DSILOG and DSITRACE or by the TASKIDS that are using NetView sequential log services. It must follow the DSTINIT statements in the member. The precoded members used in SAMPLIB are DSILOGBK and DSITRCBK.

```
[label ]          LOGINIT          [ AUTOFLIP = YES|NO ]
                                   [ ,RESUME = YES|NO ]
```

label

is the optional label for the LOGINIT statement. This label identifies the statement in any related error messages.

AUTOFLIP = YES|NO

defines whether logging should continue on the primary data set when the secondary data set is full. The value of this operand must be one of the following:

YES continues writing at the beginning of the primary data set when the secondary data set is full.

NO does not continue writing on the primary data set when the secondary data set is full.

RESUME = YES|NO

defines whether logging should begin where it left off in either data set, or logging should start at the beginning of the primary data set. The value of this operand must be one of the following:

YES resumes writing following the last entry recorded in the log data sets.

Notes:

1. RESUME=YES is not supported on VM.
2. If you define RESUME=YES and AUTOFLIP=NO, you might have to restart the network log or trace task due to full data sets, depending on where logging left off.

NO begins writing at the beginning of the primary log.

Related Statements: TASK

LU Statement

The LU statement translates LU names in interconnected networks. It assigns alias names by which logical unit names in other networks are known in this network. This statement is coded in the members of DSIPARM specified on the ALIASMEM statement. The precoded member used in SAMPLIB is DSIALTAB.

```
[label] LU targetname,targetnet,origname[,sscpname]
```

targetname

is the real name of the logical unit as it is defined to the VTAM in this domain.

targetnet

is the 1- to 8-character name of the network where the logical unit is known by its real name.

origname

is the alias translation name for this logical unit.

sscpname

is the 1- to 8-character name of the SSCP that owns the real logical unit in the *targetnet* network. *sscpname* is optional. If you do not code *sscpname*, the SSCP identifier is taken from the VTAM CDRSC definition statement.

Related Statements: ALIASMEM, ORIGNET

MAPSESS Statement

You can define a series of MAPSESS statements to describe sessions that are mapped into each performance or keep class. A session is mapped into a performance or keep class when the configuration of that session matches all the parameters of a MAPSESS statement. A session can match more than one MAPSESS statement. If it does, it is mapped by the first matching MAPSESS statement.

For interconnected networks, alias names are used for resources that are not in your local network. If you want to map resources not in your local network, you must specify the alias names instead of the real names on your MAPSESS statements.

For sessions where ER, VR, or TP data are not available to NetView, the session is assumed to match these parameters.

This sequence of statements is in the member of DSIPARM named on the INITMOD AAUINLDM PERFMEM=*membername* or KEEPMEM=*membername* definition statement.

```
[label ]      MAPSESS      PCLASS = cname
                                     [,PRI = pname ]
                                     [,SEC = sname ]
                                     [,ER = n ]
                                     [,VR = n ]
                                     [,TP = n ]

[label ]      MAPSESS      KCLASS = cname
                                     [,PRI = pname ]
                                     [,SEC = sname ]
                                     [,ER = n ]
                                     [,VR = n ]
                                     [,TP = n ]
```

label

identifies the MAPSESS statement (optional) in any related error messages.

PCLASS = cname

names the performance class, as specified on the PCLASS statement, to which sessions that match all the other MAPSESS parameters are assigned. This parameter is required.

KCLASS = cname

names the keep class, as specified on the KCLASS statement, to which sessions that match all the other MAPSESS parameters are assigned. This parameter is required.

PRI = pname

is the name of the primary end point of any session that matches this MAPSESS statement. If the desired primary end point is the SSCP, use the name specified on the SSCPNAME start option for VTAM. If you have not specified an SSCPNAME name, use VTAM for the name of the SSCP. If the PRI operand is omitted, or if a PRI=* is coded, all primary end point names match this MAPSESS statement.

SEC = *sname*

names the secondary end point of any session that matches this MAPSESS statement. If the desired secondary end point is a cross-domain SSCP, specify the CDRM name for this SSCP. If the parameter is omitted, or if SEC=* is coded, all secondary end point names match this operand.

Note: The PRI and SEC names can be coded with special characters so that a group of primary or secondary end points match the same MAPSESS statement.

To vary one character, use a question mark (“?”). For example, A?B matches any name that begins with A, ends with B, and has one character between, such as AAB, ABB, AXB, and so on. A??B matches any name that begins with A, ends with B, and has any two characters between, such as AAXB. A character must always appear in the position of a “?”; that is, A?B would not be matched by AB because there is no character replacing the “?”.

To vary a string of characters at the end of a group of resources, use an asterisk (*). For example, TSO* matches any name that begins with the letters TSO, such as TSOXYZ, TSOB2219, and so on. The * can be used only at the end of a character string. It cannot be used between characters.

ER = *n*

is the explicit route number required for a session to match this MAPSESS statement. If a value is coded, it must be from 0 through 7. If the parameter is omitted, or if ER=* is coded, all ER values match this operand.

VR = *n*

is the virtual route number required for a session to match this MAPSESS statement. If a value is coded, it must be from 0 through 7. If the parameter is omitted, or if VR=* is coded, all VR values match this operand.

TP = *n*

is the transmission priority required for a session to match this MAPSESS statement. If a value is coded, it must be from 0 through 2. If the parameter is omitted, or if TP=* is coded, all TP values match this operand.

Related Statements: KCLASS, PCLASS

MAXABEND Statement

The MAXABEND statement specifies the number of times a given NetView operator terminal session or automation task can end abnormally and be reinstated afterward by the operator. When the limit is exceeded for an operator terminal session, a second NetView operator must use either a MOVE or START command to recover the session. The number of abnormal endings increases each time the operator enters the RESET IMMED command. Issue a START, STOP, or MOVE command to reset the current count to 0. This statement is coded in the DSIDMN member of DSIPARM.

Note: The MAXABEND statement does not affect optional tasks.

```
[label]      MAXABEND      nnnnn|2
```

nnnnn|2

specifies the number of abnormal terminations allowed. You can set MAXABEND from 0 to 32767. If you do not code MAXABEND, the default is 2.

Related Statements: None

MAXLOGON Statement

The MAXLOGON statement shows the maximum number of times NetView allows an operator to recover from an incorrect logon and try again before the session with that terminal is ended. If the logon recovery limit is reached, the operator must log on through VTAM. This statement is coded in the DSIDMN member of DSIPARM.

Note: MAXLOGON does not control the number of times an operator can recover from an error in the NEW PASSWORD fields when the operator's password is changed. There is no limit on the number of times an operator can recover from an error in the NEW PASSWORD fields.

```
[label]      MAXLOGON      nnnnn|3
```

nnnnn|3

is the number of logon attempts allowed before the session is ended. You can set MAXLOGON to any number from 0 to 32767. If you do not code MAXLOGON, the default is 3.

Related Statements: None

MAXSPAN Statement

The MAXSPAN statement tells NetView the number of span names allowed. NetView sets aside storage in a table for each span. It is used in the DSIDMN member of DSIPARM.

```
[label ]      MAXSPAN      nnnnn|0
```

nnnnn|0

specifies the maximum number of spans that NetView allows. This number should be a multiple of 8. If it is not, NetView will round MAXSPAN up to the next multiple of 8. You can set MAXSPAN to any number from 0 to 32760. If you do not code MAXSPAN, it defaults to 0.

MAXSPAN should be at least as large as the number of different span names on the SPANLIST operand definitions. If you do not make MAXSPAN great enough to cover all spans, some of your span definitions might be ignored. However, to save storage, do not make the MAXSPAN number greater than you need.

Related Statements: ISPAN, SPAN, SPANLIST

For more information concerning VTAM installation, refer to "Minor Node Definitions" on page 63.

Minor Node Definitions

You can limit specific minor node resource names with span of control by coding the SPAN operand in the following VTAM and NCP definition statements:

- VTAM statements: APPL, PU, LU, LOCAL, CDRM, CDRSC, GROUP, LINE, CLUSTER. To learn how to code these VTAM definition statements, see *VTAM Installation and Resource Definition*.
- NCP statements: GROUP, LINE, CLUSTER, TERMINAL, COMP, VTERM, PU, LU. To learn how to code these NCP definition statements, see *NCP-SSP Resource Definition Reference*.

These statements are used in the member of DSIVTAM that has the same name as the associated major node.

<i>minornodename</i>	<i>statement</i>	<i>operands</i>
		SPAN=(spanname [...])

minornodename

specifies the minor node name for this resource.

statement

identifies the VTAM or NCP statement that defines the resource.

operands

are other operands for VTAM or NCP or both.

SPAN=(spanname [...])

specifies the span or spans that are to be included in the minor node resource.

The number of spans defined with the SPAN operand and the operands of SPANLIST must not exceed the value of the MAXSPAN statement. Any excess spans are ignored.

Note: To add span for minor node names, code a SPANLIST statement in NetView for the associated major node name. Once you define the span of control, there is no association between the major and minor nodes. They can be in different spans, or the major node may not be span-checked at all. If the major and minor nodes are both in the same spans, you must list the spans on both the SPANLIST statement and on the VTAM SPAN operand.

Related Statements: MAXSPAN, SPAN, SPANLIST

MODE Statement

The MODE statement defines equivalent names for duplicate logon mode names in interconnected networks. This statement is coded in the members of DSIPARM specified on the ALIASMEM statement. The precoded member used in SAMPLIB is DSIALTAB.

```
[label]      MODE      targname,targnet,origname
```

targname

is the logon mode name that applies to the target network. This name is also called the equivalent name because the MODE name becomes equivalent to the MODE name in the origin network.

targnet

is the 1- to 8-character network name of the network where the logon mode is known by its equivalent name.

origname

is the name the logon mode name is known by in the origin network.

Related Statements: ALIASMEM, ORIGNET

MSGCMD Statement

The NetView message table definition used by the command facility has changed since NetView Release 1. Instead of MSGCMD statements, in NetView Release 3, IF-THEN automation statements are required to define messages that drive command lists. A utility program is provided with the installation samples to convert NetView Release 1 MSGCMD statements to the format required for NetView Release 3. For information about how to migrate NetView Release 1 MSGCMD statements to NetView Release 3 IF-THEN automation statements, see *NetView Installation and Administration Guide*. For more information about automating messages, see *Console Automation Using NetView: Planning* and *Console Automation Using NetView: Implementing*.

Related Statements: IF-THEN

NCCFIC Statement

The NCCFIC statement names a command or command list to be run automatically when NetView is initialized. Any messages from the command or command list are sent to the system operator's console. This statement is coded in the DSIDMN member of DSIPARM.

```
[label]      NCCFIC      IC=text
```

IC=text

specifies the command or command list to run automatically when NetView is started. The command or command list must be able to run under the primary POI task (PPT).

For a list of commands that cannot be used with the PPT, see *NetView Customization: Writing Command Lists*.

Note: The default NCCFIC statement coded in the sample DSIDMN shipped with NetView is as follows:

```
NCCFIC IC=CNME1034 DSIMSG01
```

This invokes command list CNME1034 with the parameter DSIMSG01. The parameter DSIMSG01 is the name of the message automation table that will be in effect when NetView is initialized. If you wish to use another table, change the DSIMSG01 parameter to the name of the table you want to use.

Related Statements: None

NCCFID Statement

The NCCFID statement specifies the application program identification of the NetView in this domain. This statement is coded in the DSIDMN member of DSIPARM.

```
[label]      NCCFID      DOMAINID=domainid
                [ ,DMNPSW=password ]
                [ ,SUPPCHAR=suppressionchar ]
                [ ,DROP=YES|NO ]
```

DOMAINID=domainid

specifies the 1- to 5-character application program name to be used with a VTAM ACB for this NetView. This name should be unique throughout the network.

For interconnected networks, all NetViews and NCCFs in session with each other must have unique identifiers that cannot be alias names.

DMNPSW=password

specifies a password to be used for this NetView domain. If you do not code a password, the password becomes the same as the NetView identifier.

SUPPCHAR=suppressionchar

specifies a single character to be used by a NetView operator to prevent NetView from writing the command out on the terminal, hard-copy log, and network log. If this operand is not specified, NetView does not check for a suppression character.

Do not use the following characters as suppression characters:

A	equal sign (=)
B	blank
C	period (.)
D	asterisk (*)
percent (%)	ampersand (&)
comma (,)	minus (-)

DROP = YES|NO

specifies whether NetView operator sessions end under certain conditions. For example, if you specify DROP=YES and an operator who is logged on to a non-SNA remote device powers off and then on, NetView would end the operator session.

YES tells NetView to end the session. YES is the default.

NO tells NetView to attempt session recovery.

Related Statements: None

O MONIT Statement

The **O MONIT** statement specifies whether status monitor tries to reactivate inactive nodes (other than major nodes, applications, and cross-domain resources) if an unsolicited VTAM IST105I message is received by NetView. Nodes can be excluded from automatic reactivation as follows:

- By flagging them with the **NOMONIT** operand on the **STATOPT** statement
- By using the **MONOFF** command or the **MONIT STOP** command to add them to the node reactivation exclusion list.

The **O MONIT** option control statement in member **DSICNM** of **DSIPARM** is shown below.



O MONIT

O MONIT

controls automatic node reactivation.

Related Statements: **STATOPT**

OPCLASS Statement

The OPCLASS statement defines the scope of commands for a NetView operator. Operators with OPCLASS in their profile can run only commands, command operands, and command lists that are defined with these scope classes or that are not restricted by scope of commands. If you do not code OPCLASS, or if you have OPTIONS VERIFY=MINIMAL coded in DSIDMN, the operator can run any command.

NetView will not allow an operator to log on if the scope class specified on the OPCLASS statement is greater than the maximum scope class specified on all CMDCLASS, KEYCLASS, and VALCLASS statements.

This statement is coded in the member of DSIPRF specified by the PROFILEN statement associated with the operator. Refer to "OPERATOR Statement" on page 70 and "PROFILEN Statement" on page 80 to determine how a PROFILEN statement is associated with an operator. The precoded member used in SAMPLIB is DSIPROFB.

```
[label]      OPCLASS      scopeclass [...]
```

scopeclass [...]

is a number from 1 to 2040 that represents a scope class. Start your scope classes from 1 and work up to save storage space.

Related Statements: CMDCLASS, KEYCLASS, OPERATOR, OPTIONS, PROFILEN, VALCLASS

OPERATOR Statement

The OPERATOR statement identifies each operator who can log on to this NetView or who can start a session with this NetView from a NetView in another domain. It is also used to define operator identifiers that can be started as automation tasks by the AUTOTASK command. The OPERATOR statement must come before its associated PROFILEN statements. This statement is coded in the DSIOPF member of DSIPARM.

```
opid          OPERATOR          PASSWORD=password
```

opid

is the 1- to 8-character value that identifies an operator. The operator identifier must be made up of the characters A-Z, the numbers 0-9 or the special characters #, @ or \$. Each operator must have a unique operator identifier. Also, you must code an operator statement for each operator identifier you want to use for an automation task. Do not use the names of hard-copy logs, terminals, or task identifiers as operator identifiers. The following words are reserved by NetView and cannot be used as operator identifiers:

ALL, DPR, DST, HCL, LOG,
MNT, NNT, OST, PPT, SYSOP, TCT.

PASSWORD=*password*

is the 1- to 8-character operator password. You must code a password, but the password is ignored if OPTIONS VERIFY=MINIMAL is coded in DSIDMN. The password is also ignored if this operator identifier is used when starting an autotask using the AUTOTASK command. For operator identifiers set up specifically for autotasks, you can use the password to identify the operator as such.

Related Statements: OPTIONS, PROFILEN

OPTIONS Statement

The OPTIONS statement describes the type of checking NetView does at logon. It is coded in the DSIDMN member of DSIPARM. If there is no OPTIONS statement in DSIDMN, VERIFY=NORMAL is assumed.

```
[label] OPTIONS VERIFY=MINIMAL|NORMAL|MAXIMUM
```

VERIFY=MINIMAL|NORMAL|MAXIMUM

defines the level of security that NetView checks at logon.

MINIMAL ensures that the operator is defined to NetView and the operator is not already logged on. The password is not checked. The logon profile and any other logon parameters are ignored. Scope of commands and span of control are not used. The operator can enter any command and control all resources.

NORMAL checks all logon parameters, including operator identification, password, hard-copy log name, and profile. The profile, which contains information about what the operator is allowed to do, lets you limit the commands and resources that the operator can use.

MAXIMUM uses security software to check the operator identification and password. NetView checks the hard-copy log name and profile. This operand replaces the RACF operand used by NetView Release 2. However, NetView Release 3 supports RACF for compatibility purposes.

Note: You can use MAXIMUM with NetView installed on an MVS/XA operating system only. Also, you must have some form of security software installed (for example, RACF).

Related Statements: None

ORIGNET Statement

The ORIGNET statement defines the name of the network to which the LU, COS, and MODE statements apply. This statement is coded in the members of DSIPARM specified on the ALIASMEM statement. The precoded member in SAMPLIB is DSIALTAB.

```
[label] ORIGNET networkname
```

networkname

is a 1- to 8-character name of the network requesting the translation.

Related Statements: ALIASMEM, COS, LU, MODE

PARMSYN Statement

The PARMSYN statement defines new names for NetView command operands. This statement is coded in the DSICMD member of DSIPARM.

When help is used, only the original operand name is used in the help information.

All PARMSYN statements for a CMDMDL must follow the CMDMDL and precede the CMDSYN statements.

Note: Do not use PARMSYN for VTAM commands.

```
[label ]      PARMSYN      regularvalue,additionalvalue
```

regularvalue

is the original NetView command operand name.

additionalvalue

is a new 1- to 8-character operand name. NetView accepts either the old or the new operand name.

Related Statements: CMDMDL, CMDSYN

PCLASS Statement

The PCLASS statement controls the measurement parameters used by the response time monitor. These measurement parameters include:

- The time boundaries between the response time counters
- The response time definition used for a session
- The response time objective used for a session
- Whether you can display the response time of the last transaction for their session.

If you do not code any of the statements discussed in this section, the following defaults are used for all sessions:

- The boundaries between the counters are at 1, 2, 5, and 10 seconds.
- Response time is defined as "time to first character."
- You cannot display your own response time.
- No response time objective is defined.

If the defaults are acceptable, you can omit this statement, but you must not code the INITMOD AAUINLDM PERFMEM=*membername* statement.

Two statements control the response time monitor. The first, PCLASS, is used to define performance classes; the second, MAPSESS, is used to map each session into one of the defined performance classes. A performance class determines the response time measurement parameters for the sessions mapped into that performance class.

If you code PCLASS and MAPSESS statements, create a member in DSIPARM and put them in that member. The name of this new member must match the name coded on the INITMOD AAUINLDM PERFMEM=*membername* statement.

You must code one PCLASS definition statement for each performance class you define. These PCLASS statements must be the first statements that appear in the member of DSIPARM named on the INITMOD AAUINLDM PERFMEM=*membername* definition statement.

```

cname      PCLASS      [BOUNDS=(a,b,c,d)|(1,2,5,10) ]
                                     [,RTDEF=FIRST|CDEB|KEYBD|LAST ]
                                     [,DSPLYLOC=YES|NO]
                                     [,OBJTIME=mm:ss.s|00:00.0 ]
                                     [,OBJPCT=ppp|0 ]

```

cname

is the name of the performance class you define. This name is used on one or more MAPSESS statements as the value of *cname* in the PCLASS parameter. The name starts in column one of the statement and is required.

BOUNDS = (a,b,c,d)|(1,2,5,10)

lists the upper time limit on each of the first four response time counters. These times can be specified in minutes, seconds, and tenths of seconds (mm:ss.s), in seconds only (ss or ss.s), in tenths of seconds only (.s), or in minutes only (mm:). The maximum time you can specify is 1800 seconds; any value of 60 seconds or more is converted to minutes for display.

The first counter is incremented when response time is less than or equal to *a*; the second counter is incremented when response time is greater than *a* but less than or equal to *b*; and so on. The fifth response time counter is incremented when response time is greater than *d*.

If a BOUNDS parameter is specified with less than four boundaries coded, the maximum time allowed is used for the omitted boundaries. If the BOUNDS parameter is omitted, the default is BOUNDS=(1,2,5,10).

RTDEF = FIRST|CDEB|KEYBD|LAST

defines how response time is measured. It is measured as the time from the activation of a transaction, such as when the ENTER key is pressed, until a response is received. This parameter allows you to define "received" in one of the following ways.

- FIRST** means the first character of the reply from the host has arrived at the user's terminal. If RTDEF is omitted, FIRST is the default.
- CDEB** means an "SNA Change Direction" or an "SNA End Bracket" has been received at the user's terminal.
- KEYBD** means the user's terminal keyboard is unlocked.
- LAST** means the last character of the reply from the host has been received at the user's terminal.

Note: If a response time monitor is monitoring an LU to LU session and the PCLASS statement defining the session uses an RTDEF value not supported by the response time monitor, then the PCLASS statement is ignored and the session is mapped to another class. If no other classes are defined or match, the default performance class is used.

DSPLYLOC = YES|NO

controls the display of response time.

- YES** allows the terminal user to display the response time of the last transaction at the terminal.
- NO** no response is displayed.

Note: The default is set by the RTMDISP initialization statement. If RTMDISP is not specified, the default for DSPLYLOC is NO.

OBJTIME = mm:ss.s|00:00.0

is the time threshold of the performance objective. It can be entered as minutes, seconds, and tenths of seconds (mm:ss.s), in seconds only (ss or ss.s), in tenths of seconds only (.s), or in minutes only (mm:). The maximum time is 1800 seconds; any value of 60 seconds or more is converted to minutes for display.

Note: If OBJTIME is 0, no response time objective data is displayed for sessions mapped to this PCLASS statement. Refer to the MAPSESS statement to determine how sessions are mapped to a PCLASS statement.

If OBJTIME is not equal to one of the values of the BOUNDS parameter, it is rounded off to the nearest BOUNDS value.

PCLASS

Note: A response time objective includes both a time threshold and a percentage. For example, for a particular terminal you might want response times to be less than 5 seconds (the threshold) for 80% (the percentage) of the activity.

OBJPCT = *ppp*0

is the percentage portion of the performance objective.

ppp is the percentage of transactions that should take less time than the time specified by OBJTIME.

0 is the default value.

Note: If OBJPCT is 0, no response time objective data is displayed for sessions mapped to this PCLASS statement. Refer to the MAPSESS statement to determine how sessions are mapped to a PCLASS statement.

Related Statements: INITMOD, MAPSESS

POS Statement

The POS statement defines the terminals that can log on directly to this NetView. You can code either the POS or the POSPOOL definition statement, or code both. This statement is coded in the DSIDMN member of DSIPARM.

```
[label ]      POS      terminalname [...]
```

terminalname [...]

is the name of a terminal.

Use the POS statement as often as necessary to define all operator terminals.

Related Statements: POSPOOL

POSPool Statement

The POSPOOL statement defines the maximum number of terminals, that were not previously defined with the POS statements, that can log on at one time. You can code either the POSPOOL or the POS definition statement, or code both. This statement is in the DSIDMN member of DSIPARM.

```
[label] POSPOOL n|0
```

n|0

specifies the maximum number of operator terminals that can log on directly to this NetView and remain in session at the same time. Be careful when assigning this value because NetView allocates storage to accommodate each potential session. The default is 0.

Note: The maximum number coded in this statement should not include any operator identifiers reserved for automation tasks, because automation tasks do not use terminals.

Related Statements: POS

PROFILE Statement

The PROFILE statement defines the profile name to the system. PROFILE must be the first statement in each profile definition. This statement is coded in a member of DSIPRF specified by a PROFILEN statement associated with the operator. Refer to "OPERATOR Statement" on page 70 and "PROFILEN Statement" on page 80 to determine how a PROFILEN statement is associated with an operator. The precoded members used in SAMPLIB are DSIPROFA and DSIPROFB.

```

profilename          PROFILE          [ HCL = hclname ]
                               [ ,IC = text ]

```

profilename

is the name of the member that contains the profile.

HCL = hclname

is the name of the hard-copy printer that is automatically started when this operator logs on. This name must also be defined in the VTAM definition and in the NetView HARDCOPY definition statement in the DSIDMN member of DSIPARM. HCL is an optional operand; if you specify both HCL and IC, HCL must precede the IC operand and you must separate the operands with a comma.

Although each operator can be assigned to only one hard-copy printer, several operators can share the same one. However, if too many operators share the same hard-copy printer, messages for that device can accumulate and messages might not be printed for some time after they are received.

IC = text

is the command or command list that is run immediately after the logon is successful. All the text on the remainder of the statement through column 71 is treated as the text of the command, including embedded blanks and commas. No enclosing quotation marks are allowed. Only commands defined on a CMDMDL statement as regular (R) or both (B) are allowed.

Related Statements: HARDCOPY, OPERATOR, PROFILEN

PROFILEN Statement

The PROFILEN statement associates the name of a particular profile or list of profiles with an operator identification. Code PROFILEN as often as necessary to ensure that all the possible profile names are associated with a particular operator identification. An OPERATOR statement must precede each PROFILEN statement or group of statements. This statement is coded in the DSIOPF member of DSIPARM.

```
[label ]      PROFILEN      profilename [...]
```

profilename [...]

is the profile name to be associated with the operator identification defined in the preceding OPERATOR statement. The profile name is a 1- to 8-character name that matches the *profilename* given on a PROFILE statement in a profile member of DSIPRF. The first name listed in the first PROFILE statement is used by default if an operator does not specify a profile name in the logon request.

Related Statements: OPERATOR, PROFILE

R (Ratio) Statement

The R statement lets you assign initial ratio values when the first error record for a particular resource is received. These statements do not alter the wrap and ratio counts for resources already on the hardware monitor data base.

The R statement must follow the CTL statement, which specifies how many R and W statements there are.

To change the wrap and ratio values for existing resources, use the explicit SWRAP and SRATIO commands (see *NetView Operation*). The altered values remain in effect until one of the following occurs:

- The hardware alerts data base is reinitialized.
- All hardware alert records for the particular resource and all attached resources are purged.
- The values are changed by another SWRAP or SRATIO command.

This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is BNJDSEV. The precoded member used in SAMPLIB is BNJMBDST.

```
R           rtype rname vvv
```

rtype

is the resource type. This resource type must be one of the following:

- LINE
- CTRL
- LCTL
- LDEV.

rname

is the resource name. This name is a single name that corresponds to the resource type. If several resources have the same type and name, they are all affected. *rname* is required.

vvv

is the initial error-to-traffic (E/T) ratio specified for *rtype*. The maximum E/T ratio value that can be entered is 250 (interpreted by NetView as 25.0%). The default E/T ratio is 3.0% for all unspecified link-attached resources and 1.0% for all unspecified channel-attached communication resources.

Related Statements: CTL, TASK, W

RATE Statement

The RATE statement allows you to set the maximum rate at which events can be logged to the hardware monitor data base. The purpose of this function is to stop data base logging of repetitive events from a resource. The RATE function compares the time between an event being wrapped off the data base and the new event record being recorded on the data base. A filter is set to block the recording of events from the resource if the difference is less than the time specified on the RATE statement.

This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is BNJDSEV. The precoded member used in SAMPLIB is BNJMBDST.

Notes:

1. If you do not code a RATE statement, the rate value is set to zero. A value of zero turns off the RATE function.
2. Message BNJ045I is issued when the filter is set. When the cause of the excessive events is determined, delete the filter entry.
3. Message BNJ045I is issued only once every five minutes for a particular resource unless message BNJ045I is issued for another resource within the five minutes.
4. The recommended rate is 1 event per second for the predominate wrap count. This is 25 seconds using the default wrap count of 25. Low wrap counts can cause the filter to be set and the message to be issued erroneously.
5. The RATE function is not performed on events with a wrap count of zero or one.

RATE

mm:ss

mm:ss

identifies the number of minutes and seconds that must elapse, after an event is logged to the hardware monitor database, before another event can be logged.

Related Statement: w

REPORTS Statement

The **REPORTS** statement specifies whether the hardware monitor report logging starts at NetView initialization. A report record is built for each instance that input problem data is stored in the hardware monitor data base. These report records can be written to the SMF data set (MVS only) or to a data set defined by the user through a NetView exit. VM users must define a file and a user exit routine for storing report records. You can generate management reports from this file with the service level reporter (SLR) program product (MVS only) or with a similar program.

You can enable or disable record logging for report generation at initialization or at any time during a production run by issuing the **REPORTS** command.

Problem data blocked by the event or statistical recording filter (ESREC) does not become part of a report record. Data from input records that result in multiple records written to the data base are recorded in a single report record. The external log record formats are in Appendix A, "External Log Record Formats" on page 101.

This statement is coded in the member of DSIPARM specified by the **MEM** keyword of the **TASK** statement whose **TSKID** is **BNJDSERV**. The precoded member used in **SAMPLIB** is **BNJMBDST**.

Notes:

1. This statement must be coded after the **DSTINIT** statement that defines **BNJAPAMA** as the **DST** initialization exit.
2. To run **SLR** against SMF logs that NetView has written in a double-byte character set (DBCS), you need a release of **SLR** that supports DBCS.

REPORTS	OFF ON
----------------	---------------

OFF|ON

OFF means that record logging does not start automatically at initialization. **OFF** is the default.

ON means that record logging starts automatically when NetView is initialized.

Related Statements: **DSTINIT**, **TASK**

RRD Statement

The RRD statement shows the domains with which an operator who has global control can start a cross-domain session. The RRD statements can tell NetView the domains that contain a particular cross-domain resource. This statement is coded in the DSIDMN member of DSIPARM.

```
nccfid          RRD          [nodename[,...]]
```

nccfid

is the 1- to 5-character NetView identification of the domain where this NetView can start a cross-domain session. The value of *nccfid* should match the NetView identifier in the other domain. This domain name is required.

For interconnected networks, all NetViews in session with each other must have unique identifiers that cannot be alias names.

nodename [...]

specifies the node name of a resource. If you code this node name on an RRD statement, you do not have to state explicitly the domain identifier when you route VTAM commands.

The SSCP in the domain specified by *nccfid* must know the node name because NetView sends routed commands directly to that domain. These commands do not undergo any alias name translation by the sending SSCP.

Note: For interconnected networks, the alias name translation function allows the use of duplicate resource names in different networks. To avoid conflicts, you must define each node name only once within the set of RRD statements. To refer to a cross-domain resource with a duplicate name, state the appropriate domain name explicitly in the command.

Related Statements: NCCFID

SPAN Statement

The SPAN statement is used for span of control in VTAM. SPAN is coded in an operator profile. It defines the spans that an operator can add to span of control by using the START SPAN command. These spans are not in the operator's control at logon.

SPAN can be used as often as necessary to define all the desired span names. This statement is coded in a member of DSIPRF specified by a PROFILEN statement associated with the operator. Refer to "OPERATOR Statement" on page 70 and "PROFILEN Statement" on page 80 to determine how a PROFILEN statement is associated with an operator.

```
[label ]      SPAN      spanname[,...]
```

spanname [...]

specifies the 1- to 8-character name of the span. The same span name can be used in more than one profile. These span names are associated with VTAM resources through the NetView SPANLIST statement and SPAN operands in the VTAM definition.

Related Statements: OPERATOR, PROFILEN, SPANLIST

For more information concerning VTAM installation, refer to "Minor Node Definitions" on page 63.

SPANLIST Statement

The SPANLIST statement associates a span or list of spans with a particular VTAM major node. If a resource is in more than one span, code all spans that include this resource. You can use more than one SPANLIST statement for the same major node. This statement is coded in the DSISPN member of DSIPARM.

```
nodename          SPANLIST          [spanname[...]]
```

nodename

specifies a major node name that is known to VTAM. This is the name of a major node definition in DSIVTAM.

spanname [...]

specifies the unique 1- to 8-character name of the span containing this major node.

Related Statements: SPAN

For more information concerning VTAM installation, refer to "Minor Node Definitions" on page 63.

STATOPT Statement

The STATOPT parameter values are used to control the functions of the status monitor. These parameters are coded following the applicable VTAM node definition statements. This statement is coded in the member that defines a resource in DSIVTAM.

The STATOPT parameters enable the following:

- A specific description for the node is coded in place of the default node description.
- The node is excluded from automatic reactivation by NetView.
- The node is excluded from status monitor activity displays.
- The node, plus all the dependent lower nodes that follow, are omitted from status monitor's view of the network definition.

There are three basic types of STATOPT statements within major node definition statement members:

- A statement following an initial major node macro instruction statement (LBUILD, VBUILD, or PCCU macro instruction) that defines a descriptive name for the major node when the default is not desired.
- A statement following a VTAM or NCP macro instruction definition statement for a minor node.

```
* STATOPT = [(['nodedesc'][,NOMONIT][,NOACTY])|OMIT ]
```

It is not necessary to code a STATOPT statement for every VTAM node definition. The defaults for a node for which a STATOPT statement is not coded are given in the parameter descriptions below.

STATOPT =

is used to control major or minor node monitoring functions for the status monitor.

The rules for adding the STATOPT statement are as follows:

- The statements must be inserted directly after the related VTAM macro instruction definition. They must not be inserted between statements for the same VTAM macro instruction (between continuation statements).
- Each statement must be coded on a single record.
- Position 1 must contain an asterisk (*) to cause VTAM to treat it as a comment statement.
- STATOPT must start in position 16.
- All parameters must follow directly after the equal sign (=). Do not leave a blank after the equal sign.
- The VTAM VBUILD statement must be present in application major node definitions or the STATOPT statements are not recognized by the preprocessor.

STATOPT

- If *nodedesc* is coded with one or more of the other parameters, then it must be the first parameter. The other parameters can be coded in any order. *nodedesc* must be enclosed in single quote marks.

nodedesc is the 14-character description of any major or minor node. The description is displayed on the `DESCRPT` form of the status detail displays.

The description must not include any single quotes. If no description is coded, the following `STATOPT` default descriptions for the node types are used:

Node Group	Default Description
Application Major Node	APPL MAJ NODE
Application Minor Node	APPLICATION
CDRM Major Node	CDRM MAJ NODE
CDRM Minor Node	CDRM
CDRSC Major Node	CDRSC MAJ NODE
CDRSC Minor Node	CDRSC
Local Non-SNA Terminal Major Node	LOCAL 3270 MAJ
Local Non-SNA Terminal Minor Node	LOCAL dev-addr
Local SNA PU/LU Major Node	LCL SNA MAJOR
Local SNA PU Minor Node	LOCAL SNA PU
Local SNA LU Minor Node	LOCAL SNA LU
NCP Major Node	NCP MAJOR NODE
Line Minor Node	LINE
Remote PU Minor Node	PU
Remote LU Minor Node	LU
Remote Cluster Minor Node	CLUSTER
Remote Terminal Minor Node	TERMINAL
Channel-Channel Major Node	CTC MAJ NODE
Line Minor Node	LINE
PU Minor Node	PU
Switched Major Node	SWITCHED MAJOR
Switched PU Minor Node	SWITCHED PU
Switched LU Minor Node	SWITCHED LU

Note: `STATOPT = nodedesc` can be coded in `ATCSTRXX` to specify a host description. All other parameters coded in `ATCSTRXX` on a `STATOPT` statement are ignored. The default host description is `STATMON HOST`.

NOMONIT excludes the node from automatic reactivation. Code the `NOMONIT` parameter on a `STATOPT` statement for those nodes that you do not want to use automatic reactivation. If this parameter is not present, the node is included for automatic reactivation when monitoring is on. Do not code this option for major nodes or applications. Refer to "O MONIT Statement" on page 68 to determine how to enable monitoring.

NOACTY excludes the application node from activity recording. Code the `NOACTY` parameter on a `STATOPT` statement for the nodes that are excluded from activity displays. The default condition includes the node in the activity displays if the node is an application node.

OMIT excludes the node, plus all the dependent lower nodes, from the status monitor's view of the network definition. If this parameter is not coded, the node is included in the status monitor's view of the network. When you specify OMIT after a VTAM definition for a cluster, the cluster and the terminals defined on the cluster are omitted from status monitor's view of the network definition.

Using STATOPT to Define Major Nodes

STATOPT = *majnode*

STATOPT = *majnode* can be coded in your ATCCONXX VTAMLST member or in your CNMCONXX VTAMLST member to define major nodes to the status monitor.

majnode

is a 1 to 8 character major node which must follow directly after **STATOPT =** and must be followed by a blank. *majnode* must not be enclosed in quote marks.

majnode is the only STATOPT parameter that is recognized in ATCCONXX and CNMCONXX members. All other parameters coded in ATCCONXX or CNMCONXX on a STATOPT statement are ignored.

Related Statements: O MONIT

STORPOOL Statement (MVS Only)

The **STORPOOL** statement controls the method that is used for storage allocation. It is coded in the **DSIDMN** member of **DSIPARM**.

STORPOOL **SIZE = n|0**

SIZE = n|0

If you code a size from 1 to 7, NetView uses the **MVS/XA** storpool management services (**CPOOL**). The amount of pre-allocated storage specified to **CPOOL** by NetView is then multiplied by the value coded in the **STORPOOL** statement.

A value of 1 is recommended, because values greater than 1 can result in wasted storage. Setting **STORPOOL** to 0 does not take advantage of the **MVS/XA** storpool management services (**CPOOL**) and is not recommended.

The **STORPOOL** feature is used in storage requests for NetView tasks that satisfy the following criteria:

- The tasks must be running under **MVS/XA**
- The tasks must be authorized
- Subpools must be from 0 through 15
- Storage sizes must be from 1 through 4064
- The tasks must be running above the 16-megabyte storage line.

Related Statements: None

T (Timer) Statement

The T statement allows you to set timers that control how often NetView issues certain VTAM network status commands. The network status commands are used by status monitor. If you do not code this statement, NetView does not issue these commands. You can set the timers to one-minute intervals. This statement is coded in the DSICNM member of DSIPARM.

```
T  APPLS | PENDING          mm  comments
```

T
is the statement name and must be in column 1.

APPLS | PENDING

is the name of the timer to be set. This name must begin in column 3 of the statement.

APPLS specifies that you want to set the timer for NetView to automatically issue the VTAM command: D NET,APPLS.

PENDING specifies that you want to set the timer for NetView to automatically issue the VTAM command: D NET,PENDING.

mm
is the number of minutes between VTAM commands. This value must begin in column 18.

Notes:

1. Consider the strain that can be placed on the network before you decide the value for *mm* with either APPLS or PENDING. While setting a larger number of minutes means better network performance, it also means less accurate status information. Similarly, setting a low value decreases the level of performance in the network, but increases the accuracy of the status information. If you set *mm* to 0, the VTAM command is not issued.
2. For larger networks, specify values for *mm*, with either APPLS or PENDING, that are relatively prime (not multiples of each other).

comments

columns 20 through 80 are unused space for the statement. Code any comments you want included with the statement in columns 20 through 80.

Related Statements: None

TASK Statement

The TASK statement defines a task to NetView. This statement is coded in the DSIDMN member of DSIPARM.

```
[label]      TASK      MOD = modulename
                    ,TSKID = taskname
                    [,MEM = membername ]
                    [,PRI = n|9 ]
                    [,INIT = Y|N ]
```

MOD = modulename

is a 1- to 8-character name of the module that runs a task. This is a required operand. For NetView data services, the module name is DSIZDST.

TSKID = taskname

is a 1- to 8-character name that specifies the task. This operand is required and must be a unique name.

MEM = membername

is the user-defined initialization member name in DSIPARM that is used by this task. This task is responsible for the format and contents of this member. For IBM-supplied NetView tasks, the initialization member is processed only when MOD=DSIZDST. For more information, see "DSTINIT Statement" on page 24.

PRI = n|9

is a number that defines the dispatching priority of this task in relation to other subtasks running in this NetView. The value of *n* can be from 1 through 9. Number 1 is the highest priority you can assign, and 9 is the lowest. If no priority value or an incorrect value is specified, 9 is the default. Primary POI task is priority 0, hard-copy task is priority 2, and operator station task and NetView-NetView task are priority 4.

INIT = Y|N

specifies when the task begins.

Y specifies that NetView initialization starts the task.

N specifies that the NetView START command must be issued to start the task. N is the default value.

Related Statements: DSTINIT

TRANSTBL Statement

The TRANSTBL statement defines a character translation set to NetView. It is used in the DSIDMN member of DSIPARM.

```
[label]      TRANSTBL      MOD = DSIEBCDC|DSIKANJI|DSIKTKNA
```

MOD = DSIEBCDC|DSIKANJI|DSIKTKNA

specifies a particular load module name that contains a 1024-byte character translation set. Specify MOD=DSIEBCDC for EBCDIC support, MOD=DSIKANJI for Kanji support, or MOD=DSIKTKNA for Katakana support.

DSIEBCDC selects an 8-bit coded character set called EBCDIC. This is the default selection.

DSIKANJI selects a character set of symbols used in Japanese ideographic alphabets called Kanji.

DSIKTKNA selects a Japanese character translation set called Katakana.

Notes:

1. All devices must use the same character set for meaningful results.
2. The TRANSTBL value for the log printer program should be the same value as the value used for the NetView definition. NetView does not check these values for compatibility.
3. If you define more than one TRANSTBL statement, NetView uses the last one you entered.

Related Statements: None

VALCLASS Statement

The VALCLASS statement associates a command value with one or more scope classes. Operators having these scope classes coded in an OPCLASS statement in their profiles or who have global control (no OPCLASS statement in their profile) can run a NetView command using this value.

Code the VALCLASS statement immediately after its related KEYCLASS statement and before the next KEYCLASS or CMDMDL statement. If you want scope of commands for the value and not for the keyword, code KEYCLASS without any scope class. This statement is coded in the DSICMD member of DSIPARM.

```
value|=OTHER VALCLASS scopeclass[,...]
```

value|=OTHER

value specifies a value of the command operand defined by the previous KEYCLASS statement that is to be scope-checked.

See Appendix D, "Command Keywords and Values that can be Scope Checked" on page 139 for a list of keywords and keyword values that can be scope checked.

= OTHER lets a user check the scope for all other values that were not specifically entered for a command operand.

Note: You can code multiple VALCLASS statements; however, =OTHER can be coded only once and only on the last VALCLASS statement.

scopeclass [,...]

specifies a scope class that can use the value. The value of *scopeclass* must be a number from 1 to 2040.

Related Statements: CMDMDL, KEYCLASS, OPCLASS

VPDINIT Statement

The VPDINIT statement defines the parameters supplied to subtask VPDTASK at initialization. VPDTASK is the NetView subtask that collects and logs vital product data (VPD). This statement is coded in the member of DSIPARM specified by the MEM keyword on the TASK statement where MOD=DSIVMATK. The precoded member used in SAMPLIB is DSIVPARM.

```
VPDINIT      ACBNAME=acbname
              [,PASSWORD=password]
              [,VPDWAIT=nnn|30]
              [,VPDREQ=nnn|1]
              [,SNAPRQ=ON|OFF]
              [,VPDSTOR=nn|2]
```

ACBNAME=*acbname*

specifies the 1- to 8-character ACB name defined for subtask VPDTASK on the VTAM APPL statement. NetView uses ACBNAME to open its interface with VTAM. ACBNAME must match the ACBNAME specified in VTAMLST.

PASSWORD=*password*

specifies the 1- to 8-character password associated with ACBNAME. If you code a password on the VTAM APPL statement in VTAMLST for subtask VPDTASK, you must code the same password here.

VPDWAIT=*nnn*|30

specifies how long NetView should wait for a response to a request for VPD before timing out. Depending on the relative time between when the last time out check was made and when the VPD request is submitted, NetView will wait for a response to the request between *nnn* seconds and two times *nnn* seconds before timing out. The value of VPDWAIT must be a number from 30 to 999 seconds. The default is 30.

VPDREQ=*nnn*|1

specifies the maximum number of queued requests for VPD that VPDTASK accepts for processing. The request for VPD can be VPDCMD OWN, VPDCMD ALL, or VPDCMD DCE. The value of VPDREQ must be a number from 1 to 999. The default is 1.

SNAPRQ=ON|OFF

specifies whether the SNAP trace option should be turned on at VPDTASK initialization. The SNAP option can be turned on or off later, using the VPDCMD SNAP ON (or OFF) command. The default is OFF.

VPDSTOR=*nn*|2

specifies the number of kilobytes allocated for formatting VPD that comes back in response to a VPD request. The value of VPDSTOR must be a number from 2 to 32. The default is 2.

There is no standard way to determine the optimal value for VPDSTOR. If you receive message DWO019 (insufficient storage), increase the value until you no longer receive the message.

Related Statements: None

VSAMLSR Statement

The VSAMLSR statement tells NetView whether to build a pool of global VSAM buffers that it can use for all data services tasks. If the initialization statement, DSTINIT, specifies MACRF=LSR or MACRF=DFR, NetView uses this pool of buffers. This statement is coded in the DSIDMN member of DSIPARM.

```
[label]      VSAMLSR      bldvrpmodname
```

bldvrpmodname

is the name of a load module containing a compiled and link-edited BLDVRP macro instruction (list form) that specifies global VSAM buffers in NetView.

The following example uses the name that is coded in the IBM-supplied DSIZVLSR CSECT, which contains the BLDVRP macro instruction:

```
VSAMLSR      DSIZVLSR
```

Assemble and link-edit this module into the link library. The module name should be the same as *bldvrpmodname*.

If you add a VSAMLSR statement, you should also add DSTINIT MACRF=LSR or DSTINIT MACRF=DFR to one or more of the DST initialization members for DSTs with FUNCT=VSAM or FUNCT=BOTH.

Related Statements: DSTINIT

W (Wrap) Statement

The w statement lets you assign initial wrap values when the first error record for a particular resource is received. These statements do not alter the wrap and ratio counts for resources existing on the hardware alerts data base.

Initial wrap count values *nnn* shown on the three w statements can be specified for event (EV), statistical (ST), or alert (AL) data types. With the exception of alert data, wrap count statements must indicate the resource type (*rtype*).

The w and R statements assign initial wrap and ratio values when the first error record for a particular resource is received. The w statement must follow the CTL statement, which specifies the number of R and w statements.

To change the wrap and ratio values for existing resources, use the explicit SWRAP and SRATIO commands (see *NetView Operation*). The altered values remain in effect until one of the following occurs:

- The hardware alerts data base is reinitialized.
- All hardware alerts records for the particular resource and all attached resources are purged.
- The values are changed by another SWRAP or SRATIO command.

This statement is coded in the member of DSIPARM specified by the MEM keyword of the TASK statement whose TSKID is BNJDSERV. The precoded member used in SAMPLIB is BNJMBDST.

```

W      EV[ nnn025 ] rtype
W      ST[ nnn025 ] rtype
W      AL[ nnn100 ]

```

EV

is the event data type.

nnn025 selects the initial wrap count value, from 1 to 999, for EV. The default value is 025. To use the default, place a comma between EV and *rtype*.

rtype selects the 4-character resource type. Resource types are listed in *NetView Operation* but you are not restricted to only those listed.

Note: For resource types LAN and RING, the default wrap count is 100.

ST

is the statistical data type.

nnn025 selects the initial wrap count value, from 1 to 999, for ST. The default value is 025. To use the default, place a comma between ST and *rtype*.

rtype selects the 4-character resource type. Resource types are listed in *NetView Operation* but you are not restricted to only those listed.

AL

is the alert data type.

nnn**100** selects the initial wrap count value, from 1 to 999, for AL. The default value is 100.

- **Related Statements:** CTL, TASK

Appendixes

Appendix A. External Log Record Formats	101
Hardware Monitor External Log Record	101
Session Monitor External Log Record	108
External Record Type	108
External Log Subvector Formats	111
Appendix B. Network Asset Management Vital Product Data Descriptions and Sample Record Formats	123
Vital Product Data Descriptions	123
Answering Node Configuration Data	123
Product Data (Subvectors X'10' and X'11')	124
DCE Data for Modems (Subvector X'50')	126
DCE Data for DSUs/CSUs (Subvector X'50')	127
Link Configuration Data (Subvector X'52')	128
Sense Data (Subvector X'7D')	129
Attached Device Configuration Data (Subvector X'82')	130
Format of Records Created by Network Asset Management Sample Command Lists	131
Format of Subrecord Prefix	132
Format of Start Subrecords	132
Format of End Subrecords	133
Format of PU Hardware Subrecords	134
Format of PU Software Subrecords	134
Format of DCE Hardware Subrecords	135
Format of Timeout Subrecords	136
Format of Error Subrecords	136
Appendix C. Modem Configuration Change Form	137
Appendix D. Command Keywords and Values that can be Scope Checked ..	139
Command Facility	139
Hardware Monitor	142
Session Monitor	142
Status Monitor	143

Appendix A. External Log Record Formats

This appendix provides the various log record formats that the hardware monitor and session monitor use to write to external logs. These logs can be SMF logs on MVS or user-written logs.

This appendix is intended to describe the formats of the external log records, which are to be accessed by customer programs. It contains general-use programming interfaces, which allow the customer to write programs that use the services of the NetView program.

Note: In the following tables, a type of BinCD means binary coded decimal, which is a numeric value coded in binary format.

Hardware Monitor External Log Record

Table 3. External Log Record Header Format for Hardware Monitor

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
0	0	2	BRFRLEN	Length of record including this header	BinCD
2	2	2	BRFRSEG	Segment descriptor	BinCD
4	4	1	BRFRFLG	System indicator	Hex
5	5	1	BRFRRTY	Record type (set to decimal 37)	BinCD
6	6	4	BRFRTME	System time stamp (seconds accumulator)	Binary
10	A	4	BRFRDTE	System date stamp in 00YYDDDF format	Packed
14	E	4	BRFRSID	System ID	Char
18	12	4	BRFRWID	Subsystem ID (set to 'NETV')	Char
22	16	2	BRFRSUBT	Record subtype (set to decimal 3)	BinCD

Table 4 (Page 1 of 2). Hardware Monitor Header Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	4	BRFPRODI	Displacement from start of external log header to section	BinCD
+4	+4	2	BRFPROLN	Length of product section	BinCD
+6	+6	2	BRFPRONO	Number of product sections (0 or 1)	BinCD
+8	+8	4	BRFALLDI	Displacement of alert report from start of SMF header to section	BinCD
+12	+C	2	BRFALLLN	Length of alert section	BinCD
+14	+E	2	BRFALLNO	Number of alert sections (0 or 1)	BinCD
+16	+10	4	BRFEVTDI	Displacement of event report from start of SMF header to section	BinCD
+20	+14	2	BRFEVTLN	Length of event section	Num

Table 4 (Page 2 of 2). Hardware Monitor Header Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+22	+16	2	BRFEVTNO	Number of event sections (0 or 1)	BinCD
+24	+18	4	BRFSTADI	Displacement of statistical report from start of SMF header to section	BinCD
+28	+1C	2	BRFSTALN	Length of statistical section	BinCD
+30	+1E	2	BRFSTANO	Number of statistical sections (0 or 1)	BinCD
+32	+20	4	BRFMODDI	Displacement of LPDA-1 modem report from start of SMF header to section	BinCD
+36	+24	2	BRFMODLN	Length of LPDA-1 modem section	BinCD
+38	+26	2	BRFMODNO	Number of LPDA-1 modem sections (0 or 1)	BinCD
+40	+28	4	BRFLPDDI	Displacement of LPDA-2 report from start of SMF header	BinCD
+44	+2C	2	BRFLPDLN	Length of LPDA-2 section	BinCD
+46	+2E	2	BRFLPDNO	Number of LPDA-2 sections	BinCD
+48	+30	4	BRFLANDI	Displacement of local area network report from start of SMF record	BinCD
+52	+34	2	BRFLANLN	Length of local area network section	BinCD
+54	+36	2	BRFLANNO	Number of local area network sections	BinCD
+56	+38	4	BRFGENDI	Displacement of generic event report	BinCD
+60	+3C	2	BRFGENLN	Length of generic event section	BinCD
+62	+3E	2	BRFGENNO	Number of generic event sections (0 or 1)	BinCD

Table 5. Product Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	2	BRFSUBTY	Record subtype (set to 3)	BinCD
+2	+2	2	BRFRELVL	NetView release level (set to 3)	BinCD
+4	+4	4	BRFPRONM	Product name (set to 'NETV')	Char
+8	+8	8	BRFTIMST	Time stamp in form 00YYDDDFHHMMSS0S	Packed

Table 6 (Page 1 of 2). Alert Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	8	BRFDOMNM	Domain name	Char
+8	+8	8	BRFFLRNM	Failing resource name	Char
+16	+10	4	BRFFLRTY	Failing resource type	Char

Table 6 (Page 2 of 2). Alert Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+20	+14	8	BRFHINM(1)	Resource level name 1	Char
+28	+1C	4	BRFHITY(1)	Resource level type 1	Char
+32	+20	8	BRFHINM(2)	Resource level name 2	Char
+40	+28	4	BRFHITY(2)	Resource level type 2	Char
+44	+2C	8	BRFHINM(3)	Resource level name 3	Char
+52	+34	4	BRFHITY(3)	Resource level type 3	Char
+56	+38	8	BRFHINM(4)	Resource level name 4	Char
+64	+40	4	BRFHITY(4)	Resource level type 4	Char
+68	+44	8	BRFHINM(5)	Resource level name 5	Char
+76	+4C	4	BRFHITY(5)	Resource level type 5	Char
+80	+50	1	BRFCPL	Complex link indicator (0 = no 1 = yes)	Hex
+81	+51	1	BRFALT	Alert indicator (0 = no 1 = yes)	Hex

Table 7. Generic Event Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	1	BRFETYPE	Event type	Char
+1	+1	9	BRFPROID	Product ID	Char
+10	+A	4	BRFALTID	Alert ID number	Char
+14	+E	40	BRFDESC	Event description	Char
+54	+36	40	BRFCAUS1	First probable cause	Char
+94	+5E	8	BRFCDPTS	Probable-cause code points 2 to 5	Hex
+102	+66	2	BRFFLAGS	Generic flags	Char

Table 8 (Page 1 of 2). Event Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	1	BRFALRTT	Event type for event records (See note)	Hex
+1	+1	1	BRFGENCA	General cause for event records	Hex
+2	+2	1	BRFSPECA	Specific cause for event records	Hex
+3	+3	2	BRFBLKID	Block ID	Hex
+5	+5	1	BRFUACD	Action code	Hex
+6	+6	8	BRFUAQL1	Detail qualifier 1	Char
+14	+E	8	BRFUAQL2	Detail qualifier 2	Char
+22	+16	8	BRFUAQL3	Detail qualifier 3	Char
+30	+1E	48	BRF48TXT	Error description: probable cause	Char
+78	+4E	2	BRFDBKID	Detail block ID	Hex

Table 8 (Page 2 of 2). Event Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+80	+50	1	BRFDUACD	Detail action code	Hex
+81	+51	1	BRFNMJTY	NMVT type 00 = NMVT 0000 01 = NMVT 0001 02 = NMVT 0025 0F = Miscellaneous NMVT FF = Non-NMVT	Hex

Note: See *NetView Operation* for possible event types.

Table 9. Statistical Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	4	BRFTRFFC	Total traffic	BinCD
+4	+4	2	BRFTEMPS	Total temp errors	BinCD

Table 10 (Page 1 of 2). LPDA-1 Modem Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	1	BRFFAIL	Failure indicator X'00' = N/A X'01' = MODEM PROBE X'02' = LINE X'03' = REMOTE DEVICE X'04' = COMM (IF UNKNOWN) X'05' = MODEM INTERFACE	Hex
+1	+1	1	BRFLNKTY	Link type (BSC = X'01', SDLC = X'02')	Hex
+2	+2	1	BRFLMSIN	Local modem status indicator (includes modem speed) X'00' = INVALID X'01' = VALID	Hex
+3	+3	1	BRFRMSIN	remote modem status indicator (includes modem speed) X'00' = INVALID X'01' = VALID	Hex
+4	+4	2	BRFMODAD	Modem address (typical X'C3F7' (C7))	Char
+6	+6	6	BRFMODTP	Modem type (machine number or 'N/AV')	Char
+12	+C	4	BRFMODSP	Data rate ('FULL' or 'HALF')	Char
+16	+10	2	BRFLNQL	Modem line quality local (0-15)	BinCD
+18	+12	2	BRFHTCTL	Modem line hits local (0-63, X'FF' for N/A). This field is model dependent.	BinCD

Table 10 (Page 2 of 2). LPDA-1 Modem Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+20	+14	2	BRFLDBIN	Rec DB local indicator (N/A > < =) X'0000' = N/A X'0001' = REC > -4 X'0002' = REC < -48 X'0003' = REC LVL within limits	Hex
+22	+16	2	BRFLDBNO	Rec DB local number (signed decimal)	Hex
+24	+18	2	BRFLNQR	Modem line quality remote (0-15)	BinCD
+26	+1A	2	BRFHTCTR	Modem line hits remote (0-63, X'FF' for N/A)	BinCD
+28	+1C	2	BRFRDBIN	Rec DB remote indicator (N/A > < =)	Hex
+30	+1E	2	BRFRDBNO	Rec DB remote number (signed decimal)	BinCD

Table 11. LPDA-2 Report Format

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	1	BRFFAIND	Failure indicator 00 - N/A 01 - DCE or probe 02 - line 03 - remote device 04 - communications (if unknown) 05 - DCE interface	Hex
+1	+1	1	BRFLINK	Link type: 01-BSC 02-SDLC	Hex
+2	+2	1	BRFSENSE	Sense byte from DCE	Hex
+3	+3	2	BRFLADDR	Local DCE address (the right character indicates which link segment this DCE pair is located on, except when the local DCE is in idle state)	Char
+5	+5	2	BRFRADDR	Remote DCE address	Char
+7	+7	3	RESERVED		
+10	+A	30	BRFLOCAL	Local DCE report	Char
+40	+28	30	BRFRMT	Remote DCE report	Char

Table 12 (Page 1 of 2). Local and Remote Modem Reports

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	4	BRFTYPE	Modem type	Char
+4	+4	2	BRFMODEL	Modem model - two EBCDIC numbers	Char

Table 12 (Page 2 of 2). Local and Remote Modem Reports

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+6	+6	2	BRFLQ	Line quality: 0-15	BinCD
+8	+8	2	BRFWLQ	Worst line quality: 0-15	BinCD
+10	+A	2	BRFIMP	Impulse hit count: 0-63	BinCD
+12	+C	2	BRFRDBM	Receive level DBM	BinCD
+14	+E	2	BRFMRDBM	Minimum receive level DBM	BinCD
+16	+10	6	BRFSPEED	Modem speed - 'FULL' or 'BACKUP'	Char
+22	+16	4	BRFACTSP	Actual modem speed (in bits per second)	BinCD
+26	+2A	1	BRFCTRL	Modem control Point to point: 1-primary 2-secondary Multipoint: 3-control 4-tributary	Char
+27	+1B	1	BRFFEAIN	Features installed Bit 0 - coupler installed Bit 1 - reserved Bit 2 - integrated modem Bit 3 - fan-out installed Bits 4-7 - modem operating mode 0000 - X6 14.4_9.6 0001 - undefined 0010 - X6 14.4_14.4 0011 - undefined 0100 - X7 19.2_9.6 0101 - undefined 0110 - undefined 0111 - undefined 1000 - X5 9.6_9.6 1001 - undefined 1010 - X5+ 9.6_9.6 1011 - undefined 1100 - X4 4.8_4.8 1101 - undefined 1110 - undefined 1111 - undefined	Binary
+28	+1C	1	BRFFEAEER	Features in error Bit 0 - coupler in error Bit 1 - reserved Bit 2 - reserved Bit 3 - fan-out in error Bit 4 - reserved Bit 5 - modem in idle state Bit 6 - nonvital data lost Bit 7 - base modem error	Binary
+29	+1D	1	RESERVED	RESERVED	Char

Table 13. Local and Remote DSU/CSU Reports

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	4	BRFDTYPE	DSU/CSU type	Char
+4	+4	2	BRFDMODL	DSU/CSU model - two EBCDIC numbers	Char
+6	+6	2	BRFDLQ	Line quality: 0-14 (15 = N/AV)	BinCD
+8	+8	2	BRFDWLQ	Worst line quality: 0-14 (15 = N/AV)	BinCD
+10	+A	2	BRFDBIPC	Bipolar code errors: 0-63 NOTE: number reported = integer ((actual number - 1) * 2400 / transmit speed) + 1	BinCD
+12	+C	10	RESERVED	RESERVED	Char
+22	+16	4	BRFDACTS	Actual DSU/CSU speed (in bits per second)	BinCD
+26	+2A	1	BRFDCTRL	DSU/CSU control Point to point: X'01'-primary X'02'-secondary Multipoint: X'03'-control X'04'-tributary	Hex
+27	+1B	1	BRFDIND	Indicators Bit 0 - out of service code received Bit 1 - out of frame code received Bit 2 - DDS-initiated loopback Bit 3 - DSU/CSU failure Bit 4 - DSU/CSU in idle state Bit 5 - integrated DSU/CSU Bit 6 - reserved Bit 7 - reserved	Binary
+28	+1C	2	RESERVED	RESERVED	Char

Table 14. Local Area Network Report

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type
+0	+0	6	BRFLMADR	Local mac address	Char
+6	+6	6	BRFRMADR	Remote mac address	Char
+12	+C	18	BRFROUTI	Routing information	Char
+30	+1E	6	BRFUPADR	Mac address of upstream member	Char
+36	+24	6	BRFDNADR	Mac address of downstream member	Char

Session Monitor External Log Record

The session monitor writes to the external log under three conditions:

- If the response time data function is active
- If the network accounting and availability measurement function is active
- When the RECORD command is entered with the STRGDATA parameter.

The response time data function writes record type 39 when the COLLECT command with the LOG parameter is issued or at session end for an LU attached to a PU with the RTM feature.

The network accounting and availability measurement function writes record type 39 when a session is started, a session ends, or a RECORD command with the SESSTATS parameter is issued.

NetView writes a counter record (type 39) to the external log when a RECORD command with the STRGDATA parameter is issued by an operator or a command list.

Each record is made up of data sections preceded by an external log record header and a data descriptor block.

External Record Type

The session monitor writes one of the following type 39 records to the external log:

- RTM collection record
- Session end record
- Session start record
- Accounting and availability data collection record
- Combined session start-end record
- Bind failure record
- Init failure record
- Storage and event counters.

Each record is divided into sections. The following sections are shown in more detail in "External Log Subvector Formats" on page 111.

RTM Collection Record (Subtype X'0001')

With the response time measurement function active, NetView writes an RTM collection record to the external log. This happens whenever a COLLECT command with the LOG parameter is issued by an operator or a command list (CLIST). The four data sections of the RTM collection record are:

- Product ID section
- Session configuration data
- Session route data
- Session response time data.

Session End Record (Subtype X'0002')

When the session monitor network accounting and availability measurement function is active, NetView writes a session end record to the external log when a session ends. The five data sections of the session end record are:

- Product ID section
- Session configuration data

- Session route data
- Session response time data
- Accounting and availability data.

Note: The session response time data section is provided only if response time monitoring is active.

Session Start Record (Subtype X'0003')

When the session monitor network accounting and availability measurement function is active, NetView writes a session start record to the external log when a new session starts. The four data sections of the session start record are:

- Product ID section
- Session configuration data
- Session route data
- Accounting and availability data.

Accounting and Availability Data Collection Record (Subtype X'0004')

When the session monitor network accounting and availability measurement function is active, NetView writes an accounting and availability data collection record to the external log. This occurs whenever a RECORD command with the SESSTATS parameter is issued by an operator or a command list. The three data sections of the accounting and availability data collection record are:

- Product ID section
- Session configuration data
- Accounting and availability data.

Combined Session Start-End Record (Subtype X'0005')

When the session monitor network accounting and availability measurement function or the response time monitor function is active, NetView writes a combined session start-end record. This is written when a session ends before NetView has a chance to write the session start record for that session or when the response time monitor function is active and the network accounting and availability function is inactive. The five data sections of the combined session start-end record are:

- Product ID section
- Session configuration data
- Session route data
- Session response time data
- Accounting and availability data.

Note: The session response time data section is provided only if response time monitoring is active.

Bind Failure Record (Subtype X'0006')

When the session monitor network accounting and availability measurement function is active, NetView writes a bind failure record to the external log whenever a session setup fails during the BIND flow. The three data sections of the bind failure record are:

- Product ID section
- Session configuration data
- Session route data.

Init Failure Record (Subtype X'0007')

When the session monitor network accounting and availability measurement function is active, NetView writes an init failure record to the external log whenever a session setup fails during the INIT flow. The three data sections of the init failure record are:

- Product ID section
- Session configuration data
- Session route data.

Storage and Event Counter Record (Subtype X'0008')

NetView writes a counter record to the external log whenever a RECORD command with the STRGDATA parameter is issued by an operator or command list. The five data sections of the counter record are:

- Product ID section
- Event counter data
- Session awareness counter data
- Resource counter data
- Storage data.

External Log Subvector Formats

NetView builds the external log records by adding combinations of the following data sections to the external log record header section and the data descriptor section:

- Product ID section
- Session route data section
- Session configuration data section
- Response time data section
- Accounting and availability data section
- Event counter data section
- Session awareness counter data section
- Resource counter data section
- Storage data section.

The detailed format of each section follows.

Table 15 (Page 1 of 2). External Log Record Header Format for Session Monitor

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LOGRLENG	Record length	Binary	Internal
+2	+2	2	LOGRSEGD	Segment descriptor	Binary	Internal
+4	+4	1	LOGRSYSI	System indicator	Binary	svc 83
				Bit Meaning when set 0-4 Reserved 5 MVS/XA or MVS/370 6 VS2 7 VS1		
+5	+5	1	LOGRRECT	Record type (X'27')	Binary	Internal
+6	+6	4	LOGRTIME	Time stamp set by SMF in hundredths of a second.	EBCDIC	svc 83
+10	+A	4	LOGRDATE	Record was moved to the external log buffer on this date. The format is 00YYDDDF where F is the sign.	EBCDIC	svc 83
+14	+E	4	LOGRSYID	System identification (taken from SID parameter)	EBCDIC	SMCASID
+18	+12	4	LOGRSUBS	Subsystem ID 'NETV'	EBCDIC	Internal

Table 15 (Page 2 of 2). External Log Record Header Format for Session Monitor

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+22	+16	2	LOGRSUBT	Record subtype	Binary	Internal
				1 = RTM collection 2 = Session end 3 = Session start 4 = Accounting and availability data 5 = Combined session start-end 6 = Bind failure 7 = Init failure 8 = Storage and event counters		

Table 16 (Page 1 of 2). Data Descriptor Section for Response Time and Accounting Data Functions

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	4	LHDRPRDO	Offset of product section (Note 1)	Binary	Internal
+4	+4	2	LHDRPRDL	Length of product section	Binary	Internal
+6	+6	2	LHDRPRDN	Number of product sections (Note 2)	Binary	Internal
+8	+8	4	LHDRSESO	Offset of session configuration (Note 1) section	Binary	Internal
+12	+C	2	LHDRSESL	Length of session configuration section	Binary	Internal
+14	+E	2	LHDRSESN	Number of session configuration (Note 2) sections	Binary	Internal
+16	+10	4	LHDRRTEO	Offset of route data section (Note 1)	Binary	Internal
+20	+14	2	LHDRRTEL	Length of route data section	Binary	Internal
+22	+16	2	LHDRRTEN	Number of route data sections (Note 2)	Binary	Internal
+24	+18	4	LHDRRTMO	Offset of response time data section (Note 1)	Binary	Internal
+28	+1C	2	LHDRRTML	Length of response time data section	Binary	Internal
+30	+1E	2	LHDRRTMN	Number of response time data sections (Note 2)	Binary	Internal
+32	+20	4	LHDRACCO	Offset of accounting and availability data section (Note 1)	Binary	Internal

Table 16 (Page 2 of 2). Data Descriptor Section for Response Time and Accounting Data Functions

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+36	+24	2	LHDRACCL	Length of accounting and availability data section	Binary	Internal
+38	+26	2	LHDRACCN	Number of accounting and availability data sections (Note 2)	Binary	Internal

Notes:

1. The offset of the first section of this type. All offsets are relative to the beginning of the record.
2. The number of sections of this type in the record.

Table 17 (Page 1 of 2). Data Descriptor Section for Storage and Event Counter Data

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	4	LCNTPRDO	Offset of product section (Note 1)	Binary	Internal
+4	+4	2	LCNTPRDL	Length of product section	Binary	Internal
+6	+6	2	LCNTPRDN	Number of product sections (Note 2)	Binary	Internal
+8	+8	4	LCNTEVNO	Offset of event counters (Note 1)	Binary	Internal
+12	+C	2	LCNTEVNL	Length of event counter data section	Binary	Internal
+14	+E	2	LCNTEVNN	Number of event counter data sections (Note 2)	Binary	Internal
+16	+10	4	LCNTSAWO	Offset of SAW counters (Note 1)	Binary	Internal
+20	+14	2	LCNTSAWL	Length of SAW data section	Binary	Internal
+22	+16	2	LCNTSAWN	Number of SAW counter data sections (Note 2)	Binary	Internal
+24	+18	4	LCNTARBO	Offset of ARB counters (Note 1)	Binary	Internal
+28	+1C	2	LCNTARBL	Length of ARB data section	Binary	Internal
+30	+1E	2	LCNTARBN	Number of ARB data sections (Note 2)	Binary	Internal
+32	+20	4	LCNTSTGO	Offset of storage counters (Note 1)	Binary	Internal
+36	+24	2	LCNTSTGL	Length of STRG data section	Binary	Internal

Table 17 (Page 2 of 2). Data Descriptor Section for Storage and Event Counter Data

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+38	+26	2	LCNTSTGN	Number of STRG data sections (Note 2)	Binary	Internal

Notes:

1. The offset of the first section of this type. All offsets are relative to the beginning of the record.
2. The number of sections of this type in the record.

Table 18. Product Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LPRDSUBT	Record subtype	Binary	Internal
				1 = RTM collection 2 = Session end 3 = Session start 4 = Accounting and availability data 5 = Combined 6 = Bind failure 7 = Init failure 8 = Storage and event counters		
+2	+2	2	LPRDVERN	NetView release level (set to 1)	Binary	Internal
+4	+4	4	LPRDNAME	Product name (set to 'NETV')	EBCDIC	Internal

Note: LPRDSUBT is the same as LOGRRECT in the log record header section.

Table 19 (Page 1 of 2). Session Configuration Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LSESREVL	Revision level: X'0002'	Binary	VTAM
+2	+2	8	LSESPNAM	Primary resource name	EBCDIC	VTAM
+10	+A	8	LSESPUN	PU of primary resource	EBCDIC	VTAM
+18	+12	8	LSESPLNK	Primary link name (Note 1)	EBCDIC	VTAM
+26	+1A	8	LSESPSAP	Primary subarea PU	EBCDIC	VTAM
+34	+22	8	LSESPDOM	Primary domain name (NCCFID)	EBCDIC	Internal
+42	+2A	8	LSESSNAM	Secondary resource name	EBCDIC	VTAM

Table 19 (Page 2 of 2). Session Configuration Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+50	+32	8	LSESPUN	PU of secondary resource	EBCDIC	VTAM
+58	+3A	8	LSESSLNK	Secondary link name (Note 1)	EBCDIC	VTAM
+66	+42	8	LSESSAP	Secondary subarea PU name	EBCDIC	VTAM
+74	+4A	8	LSESSDOM	Secondary domain name (NCCFID)	EBCDIC	Internal
+82	+52	8	LSESPCLS	Performance class name	EBCDIC	Internal
+90	+5A	8	LSESCOST	Class of service name	EBCDIC	VTAM > V2R1
+98	+62	2	LSESERN	Explicit route number (Note 2)	Binary	VTAM > V2R1
+100	+64	2	LSESRERN	Reverse explicit route number (Note 2)	Binary	VTAM > V2R1
+102	+66	2	LSESVRN	Virtual route number (Note 2)	Binary	VTAM > V2R1
+104	+68	2	LSESTPF	Transmission priority (Note 2)	Binary	VTAM > V2R1
+106	+6A	8	LSESPCID	Unique session ID	Binary	VTAM
+114	+72	1	LSESTYPE	Session type	EBCDIC	VTAM
				1 = LU-LU 2 = SSCP-LU 3 = SSCP-PU 4 = SSCP-SSCP		
+115	+73	1	LSESXNET	Cross-network session (Y or N)	EBCDIC	VTAM > V2R1
+116	+74	1	LSESCODE	Bind failure or unbind reason code; see the following:	Binary	VTAM
				<ul style="list-style-type: none"> • Offset 7 into BINDF • Offset 15 into CDSESSF • Offset 1 into UNBIND. 		
				For more information, see <i>Systems Network Architecture Formats</i> .		

Notes:

1. This field contains the link name and the channel unit address name.
2. If no data is available from VTAM, the default is X'FF'.

Table 20. Route Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LRTEREVL	Revision level: X'0001'	Binary	Internal
+2	+2	2	LRTEENAM	Total number of nodes in session path	Binary	Internal
+4	+4	2	LRTEENAMT	Number of route ele- ments in LRTEETAB. This is the number of NCP v3 or later nodes in the session path.	Binary	Internal
+6	+6		LRTEETAB	Route element table		Route test

For each route element, there is a ten-byte entry with the following format:

Table 21. Ten-Byte Route Element Entry

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	8	LRTEENAM	Route element name	Binary	Internal
+8	+8	2	LRTEETGO	Transmission group (out) number	Binary	Route test

Note: LRTEETAB is an array of structures. LRTEENAM and LRTEETGO are fields in the structure.

Table 22 (Page 1 of 2). Response Time Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LRTMREVL	Revision level: X'0001'	Binary	Internal
+2	+2	8	LRTMCOLB	Collection period begin time stamp (Note 1)	Binary	Internal
+10	+A	8	LRTMCOLE	Collection period end time stamp (Note 1)	Binary	Internal
+18	+12	2	LRTMOBJP	Objective percentage (default is 0)	Binary	Internal
+20	+14	2	LRTMOBJB	Objective counter number (default is 1)	Binary	Internal
+22	+16	1	LRTMDEF	Response time defi- nition (default is F (first))	EBCDIC	Internal
+23	+17	1	LRTMOBJF	Objective met indi- cator (Y = yes, N = no)	EBCDIC	Internal
+24	+18	4	LRTMTRAN	Number of trans- actions measured	Binary	RTM feature

Table 22 (Page 2 of 2). Response Time Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+28	+1C	4	LRTMTOTT	Total response time (Note 2)	Binary	RTM feature
+32	+20	16	LRTMBNDS	Four four-byte fields containing counter boundaries (Note 2)	Binary	Internal
+48	+30	20	LRTMBKTS	Five four-byte fields with contents of counters	Binary	RTM feature
+68	+44	4	LRTMOBJT	Objective response time (Note 2)	Binary	Internal

Notes:

1. For LRTMCOLB and LRTMCOLE, the first four bytes of the time stamp are the local time in STCK format, and the last four are the conversion factor from GMT to local time. (Example: 982B5412 FFFCA5B.)
2. LRTMTOTT, LRTMBNDS, and LRTMOBJT are in tenths-of-seconds.

Table 23 (Page 1 of 2). Accounting and Availability Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	2	LACCREV L	Revision level: X'0002'	Binary	Internal
+2	+2	2		RESERVED		
+4	+4	8	LACCBEGT	Collection period begin time stamp (Note 1)	Binary	Internal
+12	+C	8	LACCENDT	Collection period end time stamp (Note 1)	Binary	Internal
+20	+14	4	LACCPBCB	Number of control PIUS sent from primary to secondary	Binary	Internal
+24	+18	4	LACCPCCC	Number of control characters sent from primary to secondary	Binary	Internal
+28	+1C	4	LACCSCBC	Number of control PIUS sent from secondary to primary	Binary	Internal
+32	+20	4	LACCSCCC	Number of control characters sent from secondary to primary	Binary	Internal
+36	+24	4	LACCP TBC	Number of text PIUS sent from primary to secondary	Binary	Internal
+40	+28	4	LACCP TCC	Number of text characters sent from primary to secondary	Binary	Internal

Table 23 (Page 2 of 2). Accounting and Availability Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+44	+2C	4	LACCSTBC	Number of text PIUS sent from secondary to primary	Binary	Internal
+48	+30	4	LACCSTCC	Number of text characters sent secondary to primary	Binary	Internal

Notes:

1. For LACCBEGT and LACCENDT, the first four bytes of the time stamp are local time in STCK format. The last four bytes are the conversion factor from GMT to local time. (Example: 982B5412 FFFCA5B.)
2. The session monitor uses the indicators in the first byte of the RH to select control and text PIUS. BSC connections do not have control PIUS.

Table 24. Event Counters Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	8	LEVNSTIM	Start of the recording period	Binary	Internal
+8	+8	8	LEVNETIM	End of the recording period	Binary	Internal
+16	+10	8	LEVNDMID	Domain ID (NCCF ID)	Binary	Internal
+24	+18	4	LEVNPIUB	PIU trace buffers processed	Binary	Internal
+28	+1C	4	LEVNPIUS	PIUS processed	Binary	Internal
+32	+20	4	LEVNSAWB	SAW buffers processed	Binary	Internal
+36	+24	4	LEVNSESS	Session start notifications	Binary	Internal
+40	+28	4	LEVNSESE	Session end notifications	Binary	Internal
+44	+2C	4	LEVNSESR	Sessions recorded to VSAM	Binary	Internal

Notes:

1. For LEVNSTIM and LEVNETIM, the first four bytes of the time stamp are local time in STCK format. The last four bytes are the conversion factor from GMT to local time. (Example: 982B5412 FFFCA5B.)
2. SAW (session awareness) is a VTAM/session monitor interface through which information about network session activity is exchanged.
3. PIU (path information unit): See *Systems Network Architecture Formats*.

Table 25. Session Awareness Counters Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	4	LSAWASBC	Number of non-filtered sessions	Binary	Internal
+4	+4	4	LSAWASBM	Session highwater mark	Binary	Internal
+8	+8	4	LSAWFLTC	Sessions being filtered	Binary	Internal
+12	+C	4	LSAWFLTM	Filter highwater mark	Binary	Internal
+16	+10	4	LSAWHSTE	Sessions with an endpoint in the host	Binary	Internal
+20	+14	4	LSAWRTMC	Sessions keeping RTM data	Binary	Internal
+24	+18	4	LSAWXNTC	Sessions keeping cross-network data	Binary	Internal
+28	+1C	4	LSAWDOMC	Sessions keeping domain data	EBCDIC	Internal
+32	+20	4	LSAWACTC	Sessions keeping accounting data	Binary	Internal
+36	+24	4	LSAWSSCP	Number of sscp to sscp sessions	Binary	Internal
+40	+28	4	LSAWSCPM	SSCP to SSCP highwater mark	Binary	Internal
+44	+2C	4	LSAWSPPU	Number of sscp to PU sessions	Binary	Internal
+48	+30	4	LSAWSPUM	SSCP TO PU highwater mark	Binary	Internal
+52	+34	4	LSAWSPLU	Number of sscp to LU sessions	Binary	Internal
+56	+38	4	LSAWSLUM	SSCP to LU highwater mark	Binary	Internal
+60	+3C	4	LSAWLULU	Number of LU to LU sessions	Binary	Internal
+64	+40	4	LSAWLULM	LU to LU highwater mark	Binary	Internal
+68	+44	4	LSAWRCDQ	Sessions waiting to be recorded	Binary	Internal
+72	+48	4	LSAWRDQM	Record queue highwater mark	Binary	Internal

Notes:

1. LSAWFLTC is the sum of the current sessions-filtered counts from session monitor and VTAM. VTAM Version 3 Release 2 Enhancements allow sessions to be filtered at VTAM; however, your SAW=NO filtering should be done at VTAM rather than NetView.
2. LSAWFLTM is the sum of the sessions-filtered highwater counts from session monitor and VTAM since the last RECORD STRGDATA request.

Table 26. Resource Counter Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	4	LARBCNT	Number of resource control blocks	Binary	Internal
+4	+4	4	LARBMAX	ARB highwater mark	Binary	Internal
+8	+8	4	LARBSSCP	Number of SSCP resource control blocks	Binary	Internal
+12	+C	4	LARBSCPM	SSCP ARB highwater mark	Binary	Internal
+16	+10	4	LARBPU	Number of PU resource control blocks	Binary	Internal
+20	+14	4	LARBPUX	PU ARB highwater mark	Binary	Internal
+24	+18	4	LARBLU	Number of LU resource control blocks	Binary	Internal
+28	+1C	4	LARBLUMX	LU ARB highwater mark	Binary	Internal
+32	+20	4	LARBLNK	Number of link/CUA ARBS	Binary	Internal
+36	+24	4	LARBLNKM	Link/CUA highwater mark	Binary	Internal

Note: The session monitor creates active resource control blocks (ARBS) for all resources involved in sessions known to the session monitor.

Table 27. Storage Counter Data Section

Offset Dec.	Offset Hex.	Length Bytes	Field Name	Description	Type	Source
+0	+0	4	LSTGRM	Storage used for RTM data	Binary	Internal
+4	+4	4	LSTGPARM	Storage used for PARM data	Binary	Internal
+8	+8	4	LSTGTRCE	Storage used for TRACE data	Binary	Internal
+12	+C	4	LSTGASB	Storage used for ASB data	Binary	Internal
+16	+10	4	LSTGARB	Storage used for ARB data	Binary	Internal
+20	+14	4	LSTGACCT	Storage used for ACCT data	Binary	Internal

Notes:

1. The RTM counter represents the number of bytes of storage used for response time monitor data.
2. The PARM counter represents the number of bytes of storage used for session parameter data.
3. The TRACE counter represents the number of bytes of storage used for session

trace data. This includes PIU trace, boundary function trace, and gateway trace.

4. The ASB counter represents the number of bytes of storage used for active session block control blocks.
5. The ARB counter represents the number of bytes of storage used for active resource block control blocks.
6. The ACCT counter represents the number of bytes of storage used for accounting data.

Appendix B. Network Asset Management Vital Product Data Descriptions and Sample Record Formats

This appendix provides information about:

- Vital product data (VPD) returned from the VPDCMD command
- The record formats used by the sample network asset management command lists.

Use the information provided in this appendix as a reference when interpreting messages returned from the VPDCMD command, when modifying the sample network asset management command lists or when writing your own network asset management command lists.

This appendix is intended to aid the customer in writing command lists to manage vital product data. It contains general-use programming interfaces, which allow the customer to write programs that use the services of the NetView program.

Vital Product Data Descriptions

Messages that are returned in response to the VPDCMD command contain the following types of VPD:

- Answering node configuration data
- Product data
- DCE data
- Link configuration data
- Sense data
- Attached device configuration data.

Different devices support different fields of the subvectors that return the VPD. A message is built using all the fields supported by an answering device. The message reflects all the VPD that is supplied by the device (including any supported fields that the device returns with values of blanks or zeros.) Some devices may return values of blanks or zeros for fields that have no meaning to the device. More information on some of the fields may be available from the applicable hardware and software publications.

Answering Node Configuration Data

Answering node configuration data describes the node that answered the VPD request. Answering node configuration data is returned in the following messages:

- DWO100I
- DWO103I.

Messages DWO100I and DWO103I contain the same fields. Message DWO103I is issued instead of message DWO100I if the configuration reported by VTAM includes more names than the routine can manage. This happens when the amount of storage allocated during initialization (on the VPDCMD operand of the VPDINIT statement) is not sufficient. Because of the storage restriction, message DWO103I does not contain information about the complete configuration—the higher level node identification is not included.

Messages DWO100I and DWO103I are two-part messages. One part of the message contains information about the node where the VPD request originated. The other part of the message shows the configuration of the originating node.

The format of messages DWO100I and DWO103I is:

```
REQID reqid : ORIG nodetype nodename CNFG nodetype nodename [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *nodetype* and *nodename* following ORIG describe the node where the VPD request originated. The *nodetype* can be one of the following:

- PU means the physical unit name
- LSN means the link station name
- LU means the logical unit name
- LINK means the link name
- CH/LINK means the channel-link name.

The *nodename* is the name of the originating node.

The pair or pairs of *nodetypes* and *nodenames* following CNFG describe the configuration of the originating node. The *nodetypes* can be one of the following:

- PU means the physical unit name
- LSN means the link station name
- LU means the logical unit name
- LINK means the link name
- CH/LINK means the channel-link name.

The *nodenames* are the names that make up the configuration of the originating node.

Product Data (Subvectors X'10' and X'11')

Message DWO102I contains data about the type of product of a particular node. The data is returned in subvectors X'10' and X'11'.

The format of message DWO102I is:

```
REQID reqid : prodid vpdfield=xxx [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *prodid* identifies the type of product. The *prodid* can have one of the following values:

- IBM-HW means IBM hardware
- MIX-HW means IBM or non-IBM hardware (not distinguished)
- OEM-HW means non-IBM hardware
- IBM-SW means IBM software
- MIX-SW means IBM or non-IBM software (not distinguished)
- OEM-SW means non-IBM software.

The *vpdfld=xxx* represents a combination of various fields and values that describe the type of product of a particular node. The message reflects all the data returned from the device. Depending on the device, different fields are used to build this portion of the message. All the fields returned from the device are used (even if they contain blanks or zeros).

The fields that can be contained in DW0102I are:

Field	Value
M/T	The machine type of the hardware product
MDL	The machine model number of the hardware product
MFG	The plant of manufacture of the hardware product
S/N	The sequence number of the hardware product
EM/T	The machine type of the emulated product
EMDL	The machine number of the emulated product
COMPID	The software serviceable component identifier
REL	The software serviceable component release level
VER	The software product common version identifier
RLS	The software product common release identifier
MOD	The software product common modification identifier
SPROD	The software common product name
NODEID	The software product customization identifier
PPN	The software program product number
CSD	The software product customization date in YY/DDD format
CST	The software product customization time in HH:MM format
ECL	The microcode EC level
HPROD	The hardware product common name
VID	The vendor identification.

DCE Data for Modems (Subvector X'50')

Message DWO112I contains DCE data for modems retrieved from a DCE that supports read-configuration LPDA-2 commands. The data is returned in subvector X'50'.

The format of message DWO112I is:

```
REQID reqid : vpdfield=xxx [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *vpdfield=xxx* represents a combination of various fields and values that contain LPDA-2 DCE link connection subsystem data. The message reflects all the data returned from the device. Depending on the device, different fields are used to build this portion of the message. All the fields returned from the device are used (even if they contain blanks or zeros). More information on some of the fields may be available in your modem publications.

The fields that can be contained in DWO112I are:

Field	Value
CLSL	The current link segment level
ADDR	The DCE address
DCESNS	The 2-character LPDA2 sense code
M/T	The modem type
MDL	The modem model
PORT	The DCE port identifier to which the DTE is connected (fan in/fan out modems only)
FEAT	The modem features. The value of FEAT will be one of the following: <ul style="list-style-type: none">• CPL means a coupler is installed• DMPX means a data multiplexer is installed• FAN means a fan-out is installed• DTE means DTE is connected• DIR means direction connection• TAI means tailed cable connection.
CECL	The card EC level
PECL	The packaging EC level
HROS	The high speed ROS EC level
LROS	The low speed ROS EC level
S/N	The serial number
SPD	The speed in use: full or backup
NSP	The normal (full) speed
BSP	The backup speed

- FNC** The network function. The value of FNC will be one of the following:
- PRI means the modem is primary in point to point
 - SEC means the modem is secondary in point to point
 - CTL means the modem is control in multipoint
 - TRI means the modem is tributary in multipoint.
- CLK** The clock options. The value of CLK will be one of the following:
- INT means the modem uses an internal clock
 - EXT means the modem uses a DTE-provided clock
 - RCV means the transmit clock is locked to the receive clock.

DCE Data for DSUs/CSUs (Subvector X'50')

Message DWO113I contains DSU/CSU data retrieved from a local DCE that supports modem-and-line status LPDA-2 commands. Message DWO114I contains DSU/CSU data retrieved from a remote DCE that supports modem-and-line status commands. The data is returned in subvector X'50'.

The format of messages DWO113I and DWO114I is:

```
REQID reqid : vpdfield=xxx [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *vpdfield=xxx* represents a combination of various fields and values that describe the node that could not satisfy a VPD request. The message reflects all the data returned from the device. Depending on the device, different fields are used to build this portion of the message. All the fields returned from the device are used (even if they contain blanks or zeros). More information on some of the fields may be available in your DSU/CSU publications.

The fields that can be contained in DWO113I and DWO114I are:

Field	Value
ADDR	The DCE address
DCESNS	The 2-character LPDA2 sense code
M/T	The DSU/CSU type
MDL	The DSU/CSU model
FEAT	The modem features. The value of FEAT will be one of the following: <ul style="list-style-type: none"> • DTE means DTE is connected • DDS means DDS is connected • DIR means direction connection • TAI means tailed cable connection.
CECL	The card EC level
MECL	The micro code EC level
S/N	The serial number

- FNC** The network function. The value of FNC will be one of the following:
- PRI means the modem is primary in point to point
 - SEC means the modem is secondary in point to point
 - CTL means the modem is control in multipoint
 - TRI means the modem is tributary in multipoint.
- XSP** The transmission speed.

Link Configuration Data (Subvector X'52')

Message DWO110I contains data about the configuration of a link connection. The data is returned in subvector X'52'.

The format of message DWO110I is:

```
REQID reqid : vpdfield=xxx [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *vpdfield=xxx* represents a combination of various fields and values that describe the link connection. The message reflects all the data returned from the device. Depending on the device, different fields are used to build this portion of the message. All the fields returned from the device are used (even if they contain blanks or zeros). More information on some of the fields may be available in your NCP publications.

The fields that can be contained in DWO110I are:

Field	Value
PRT	The port address (see <i>NCP and EP Reference Summary and Data Areas</i>)
RDLCADR	The remote device SDLC address
MDT	The modem features. The value of MDT will be one of the following: <ul style="list-style-type: none"> • STD means that the modem does not have a multiplex link feature • DMPX means that the modem has a multiplex link feature • EXT means that the local DCE is external to the sending node • ITG means that the local DCE is integrated to the sending node.
LSLN	The number of link segments active on this link connection for a given link station
LDA	The local device address
MCN	The modem correlation number

STA The LCS link station attributes. STA can be further qualified with one value from each of the following groups:

- Link station role
 - PRI means the node is the primary link station
 - SEC means the node is the secondary link station
 - NEG means the node can negotiate the role of the link station.
- Remote link station node type
 - T01 means the remote link station node type is 01
 - T02 means the remote link station node type is 02
 - T04 means the remote link station node type is 04
 - T21 means the remote link station node type is 2.1
 - NSN means the remote link station node type is non-SNA.

LKA The link attributes. LKA can be further qualified with one value from each of the following groups:

- Connection type
 - LSD means the connection is leased
 - SWC means the connection is switched.
- Connection mode
 - HDX means the connection mode is half-duplex
 - FDX means the connection mode is full-duplex.
- Connection protocol
 - SDLC means the connection protocol is SDLC
 - BSC means the connection protocol is BSC
 - S/S means the connection protocol is Start/Stop.

PTP The connection configuration is point-to-point

MTP The connection configuration is multipoint

FPD The LPDA fault LSL descriptor.

Sense Data (Subvector X'7D')

Message DWO111I contains SNA sense data supplied by a node that could not satisfy a VPD request. The data is returned in subvector X'7D'.

The format of message DWO111I is:

```
REQID reqid : SNS sensecode
```

The *reqid* is the ID that correlates this reply to a specific request.

The *sensecode* following SNS is the 8-character hexadecimal sense code of the node that could not satisfy a VPD request.

Attached Device Configuration Data (Subvector X'82')

Message DWO1011 contains data about the configuration of a device attached to the node that reports the VPD. The data is returned in subvector X'82'.

The format of message DWO1011 is:

```
REQID reqid : vpdfield=xxx [...]
```

The *reqid* is the ID that correlates this reply to a specific request.

The *vpdfield=xxx* represents a combination of various fields and values that describe the attached device. The message reflects all the data returned from the device. Depending on the device, different fields are used to build this portion of the message. All the fields returned from the device are used (even if they contain blanks or zeros).

The fields that can be contained in DWO1011 are:

Field	Value
PORT	The port number where this device is attached
PWROS	The power-on status of this device
PWROL	Specifies whether this device has been powered-on since the last time a solicitation for VPD was issued.

Format of Records Created by Network Asset Management Sample Command Lists

The sample network asset management command lists are:

Command List	Function
VPDLOGC	Generates and logs start and end records when called by the VPDALL command
VPDPU	Collects and logs VPD from a specified PU (and optionally its ports) in your domain
VPDDCE	Collects and logs VPD from all of the DCEs existing in a direct path between a specified NCP and PU in your domain
VPDXDOM	In conjunction with the VPDLOGC, VPDPU, and VPDDCE command lists and a message automation definition, the VPDXDOM command list allows a focal point to collect VPD from devices attached to another NetView program. VPDXDOM sets the record type number of the NetView soliciting for VPD to that of the focal point NetView. This ensures that all of the collected VPD is logged with the same record type number under the focal point NetView.

You can use the sample command lists as they are shipped to perform basic VPD collection. If the sample command lists do not suit your particular network asset management requirements, you can modify the existing command lists, or you can write your own.

The sample command lists assign a record type number of 37 to each log record created. Record type number 37 is the record type number for hardware monitor records. The record type number is used to log the VPD into an external file. The record type number is set as a common global variable named SMFVPD during execution of NetView's initial command list, CNME1034. If command list CNME1034 is not run, global variable SMFVPD is not set. If SMFVPD is not set, the sample command list will set it to its default (37) and continue processing.

If you plan to write your own command lists to collect and log VPD, you must do the following:

- Define the format of each type of log record you want to create.
- Assign a record type number between 128 and 255 to each type of log record created.
- Install an external logging facility if you want to log VPD into an external file. After logging data into an external file, you can manipulate the data using tools such as SLR or Info/Management.

The records written by the sample command lists through the NetView external logging facility consist of a common prefix that contains the SMF header followed by variable data. The variable data is contained in one of the following subrecords:

- Start (subtype s)
- End (subtype E)
- PU hardware (subtype P)
- PU software (subtype F)
- DCE hardware (subtype M)
- Timeout (subtype T)
- Error (subtype w).

Each subrecord contains a collect identifier field. This field correlates the records written during the same data collection. The last two digits of the collect identifier indicate the method used to collect the data.

The following sections describe the format of the common prefix and the different subrecords.

Format of Subrecord Prefix

Each record created by the sample command lists is preceded by a common record prefix, or header, which contains identifying information about the record.

Table 28. Format of Common Record Prefix

Record Offset	Offset in CMD Proc	Length Bytes	Description
0000	___	2	Record length
0002	___	2	Reserved
0004	___	1	Reserved
0005	___	1	Record number
0006	___	4	TIME (sec/100)
0010	___	4	DATE (00yyddf)
0014	___	4	System ID
0018	___	4	Subsystem ID (set to VPD)
0022	___	2	Subsystem record number (set to 22)

Format of Start Subrecords

Start subrecords (subtype s) contain information about the start of data collection. Start subrecords are written by VPDLOGC, which is called by the VPDALL command at the start of data collection.

Table 29 (Page 1 of 2). Format of Start Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (s)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mmdyyhhmm99)

Table 29 (Page 2 of 2). Format of Start Subrecords

Record Offset	Offset In CMD Proc	Length Bytes	Description
0045	0022	8	Operator ID
0053	0030	8	Req (set to VPDALL)
0061	0038	3	Trailer (set to VPD)

Format of End Subrecords

End subrecords (subtype E) contain information about the end of data collection. End subrecords are written by VPDLOGC, which is called by the VPDALL command at the end of data collection.

Table 30. Format of End Subrecords

Record Offset	Offset In CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (E)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mmdyyhhmm99)
0045	0022	8	Operator ID
0053	0030	12	End of collect (mmdyyhhmss)
0065	0042	8	Record counter (see note 1)
0073	0050	8	Number of calls to VPDP (see note 2)
0081	0058	8	Reserved
0089	0066	8	Number of calls to VPDDCE (see note 2)
0097	0074	8	Reserved
0105	0082	3	Trailer (set to VPD)

Notes:

1. This field contains the total number of records, generated since the START sub-record, that can be written to an external logging facility. This field does not represent the number of records that were successfully written to the external logging facility.
2. This field contains the total number of calls made to the VPDP or VPDDCE command list. It does not represent the number of successful completions of the command list.

Format of PU Hardware Subrecords

PU hardware subrecords (subtype P) contain information about the hardware characteristics of a PU. PU hardware subrecords are written by the VPDP command list.

Table 31. Format of PU Hardware Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (P)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mmddyhhmm99)
0045	0022	5	Machine type
0050	0027	3	Machine model
0053	0030	3	Manufacturer ID
0056	0033	7	Sequence number
0063	0040	10	EC level
0073	0050	8	LU name
0081	0058	8	PU name
0089	0066	8	Link name
0097	0074	8	PU type 4 or 5 name
0105	0082	6	Attached port number
0111	0088	1	Current power on status
0112	0089	1	Power on status since last solicitation
0113	0090	12	Reserved
0125	0102	8	Reserved
0133	0110	3	Trailer (set to VPD)
0136	0113	1 to 50	Reserved

Format of PU Software Subrecords

PU software subrecords (subtype F) contain information about the software characteristics of a PU. PU software subrecords are written by the VPDP command list.

Table 32 (Page 1 of 2). Format of PU Software Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (F)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mmddyhhmm99)
0045	0022	9	Component ID
0054	0031	3	Release level
0057	0034	6	Customization date
0063	0040	5	Customization time
0068	0045	5	Reserved

Table 32 (Page 2 of 2). Format of PU Software Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0073	0050	8	LU name
0081	0058	8	PU name
0089	0066	8	Link name
0097	0074	8	PU type 4 or 5 name
0105	0082	3	Trailer (set to VPD)

Format of DCE Hardware Subrecords

DCE hardware subrecords (subtype M) contain information about the hardware characteristics of a DCE. DCE hardware subrecords are written by the VPDDCE command list.

Table 33. Format of DCE Hardware Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (M)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mmdyyhhmm99)
0045	0022	5	Machine type
0050	0027	3	Machine model
0053	0030	7	Serial number
0060	0037	1	Card EC level
0061	0038	1	Packaging EC level micro code EC level
0062	0039	11	Reserved
0073	0050	8	Reserved
0081	0058	8	PU name
0089	0066	8	Link name
0097	0074	8	PU type 4 or 5 name
0105	0082	1	Total link segment level
0106	0083	1	Current link segment level
0107	0084	2	DCE address
0109	0086	1	DCE position (see note 1)
0110	0087	3	Trailer (set to VPD)
0113	0090	20	Link station attribute (see note 2)
0133	0110	Varied	DCE features (see note 2)

Notes:

1. This field contains a character that identifies the sequence in which the DCEs are connected. A 1 in this field indicates the DCE is closest to the NCP. An N in this field indicates the DCE is furthest from the NCP.
2. These fields can be defined by the user.

Format of Timeout Subrecords

Timeout subrecords (subtype T) contain information about a VPD request that timed out before it was completed. Timeout subrecords are written by the VPDP or VPDDCE command lists when the ERROR option is specified.

Table 34. Format of Timeout Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (T)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mddyhhmm99)
0045	0022	8	Operator ID
0053	0030	8	Request specified
0061	0038	8	Node name 1
0069	0046	8	Node name 2
0077	0054	3	Trailer (set to VPD)

Format of Error Subrecords

Error subrecords (subtype W) contain information about a VPD request that failed. Error subrecords are written by the VPDP or VPDDCE command lists when the ERROR option is specified.

Table 35. Format of Error Subrecords

Record Offset	Offset in CMD Proc	Length Bytes	Description
0024	0001	1	Record subtype (W)
0025	0002	8	NetView domain ID
0033	0010	12	Collect identifier (mddyhhmm99)
0045	0022	8	Operator ID
0053	0030	8	Request specified
0061	0038	8	Node name 1
0069	0046	8	Node name 2
0077	0054	8	Error message number
0085	0062	3	Trailer (set to VPD)

Appendix C. Modem Configuration Change Form

This appendix contains a one-page form that system programmers can use to give modem configuration information to operators. The fields on the form correspond to fields on the MDMCNFG panels. After the operators receive the completed form, they can enter the information using the MDMCNFG command.

The form instructs the operator not to change any fields that are not filled in on the form.

Modem Configuration Change Form

Operators - Do not change entries that have been left blank on this form.

Modem Address:

ID _____ (Using Node Network Name)
Station _____ (Station Network Name)
Modem _____ (Local, Remote)
Level _____ (1-2)

Modem Configuration Changes:

Speed Control Mode _____ (M = Modem, D = DTE)
Training Sequence _____ (L = Long, S = Short)
Configuration _____ (M = Multipoint, P = Point-to-Point)
Network Function _____ (C = Control/Primary, S = Secondary)
Antistreaming _____ (Y = Yes, N = No)
Transmit Clock Option _____ (I = Internal, E = External, R = Receive)
Complementary RFS Delay _____ ms (0 to 250 in 10ms increments)
Default Speed _____ (F = Full, B = Backup)
Local Loop Back Wrap _____ (Y = Yes, N = No)
Customer Information _____ (10-Character Limit)
Receive Level Threshold _____ (-43 to 0)
Impulse Hits Threshold _____ (0 to 63)
Line Quality Threshold _____ (0 to 14)
Auto-Answer _____ (Y = Yes, N = No)
Dialing _____ (T = Tone, P = Pulse)
First Phone Number _____
Second Phone Number _____ (Valid Digits: 0-9, P = Pulse)

Appendix D. Command Keywords and Values that can be Scope Checked

This appendix contains a table for each component of the NetView program. The tables list the commands that have keywords that can be scope checked and any scope checked values associated with that keyword.

Notes:

1. You can scope check all commands and command lists coded in DSICMD. This appendix lists only the commands having keywords that can be scope checked.
2. You cannot scope check command list parameters.

Command Facility

Command	Keyword	Value
AFTER	PPT SAVE	
AINQ	ORIGNAME TARGNAME TYPE	name name LU COS MODE
AT	PPT SAVE	
AUPD	ORIGNET	name
AUTOMSG	MEMBER OFF STATUS	membername
AUTOTASK	OPID CONSOLE DROP	operid nn
BGNSESS	APPLID SRCLU	applid srclu
CANCMD	TAG ID SP	tag name spname
CCPDR	name	
CCPLOADF	name	
CCPLOADI	name	
CCPLOADT	name	
DEFAULTS	HOLD BEEP SYSLOG NETLOG HCYLOG DISPLAY CMD	ENABLE DISABLE ENABLE DISABLE YES NO YES NO YES NO YES NO HI LOW

Command	Keyword	Value
DISPCMD	ALL	name
	ID	operid
	OP	spname
	SP	
DISPCNFG	ID	name
	ALL	name
	PRODUCT	name
	LINE	name
	ADAPTER	name
	STATION	
	DISPLAY	
	VERIFY	
DUMP		
EVERY	PPT	
	SAVE	
EXCMD	opid	
	cmd	
LINESTAT	ID	name
	PORT	portnumber
	LINE	name
	HALF	
	FULL	
	SNBU	
	NOSNBU	
	SSL	
LINKDATA	SP	service-point-name
	APPL	application-name
	LINE	line-name
	RESOURCE	resource-name
	UN	using-node
	ENTRYLCC	entry-LCC
	RD	remote-device
	EXITLCC	exit-LCC
LINKPD	SP	service-point-name
	APPL	application-name
	LINE	line-name
	RESOURCE	resource-name
	UN	using-node
	ENTRYLCC	entry-LCC
	RD	remote-device
	EXITLCC	exit-LCC
LINKTEST	SP	service-point-name
	APPL	application-name
	LINE	line-name
	RESOURCE	resource-name
	UN	using-node
	ENTRYLCC	entry-LCC
	RD	remote-device
	EXITLCC	exit-LCC
SELFCNT	number-of-repetitions	
LIST	DST	dstname
	OP	ALL operid PPT '' STATUS = CNMSESS STATUS = AMLUSESS

Command	Keyword	Value
LPDA	ID	name
	STATION	name
	LINE	name
	QUERY	
	TYPE1	
	TYPE2	
	TYPE3	
	NONE ALLOW BLOCK	
MDMCNFG	ID	node
	STATION	devicename
	LEVEL	1 2
	MODEM	LOCAL REMOTE
	BROWSE	NOCHANGE CONFIG COUPLER
	CHANGE	CONFIG COUPLER
MDMCNTL	ID	usingnode
	STATION	devicename
	LEVEL	1 2
	MODEM	LOCAL REMOTE ALLRMT
	SPEED	FULL BACKUP
	CONTACT	CLOSE OPEN QUERY
	CONNECT	phone#1 (phone#1,phone#2,...)
	DISCONN	
MVS	command verb first verb function	
OVERRIDE	HOLD	ENABLE DISABLE DEFAULT
	BEEP	ENABLE DISABLE DEFAULT
	SYSLOG	YES NO DEFAULT
	NETLOG	YES NO DEFAULT
	HCYLOG	YES NO DEFAULT
	DISPLAY	YES NO DEFAULT
	CMD	HI LOW DEFAULT
PURGE	TIMER	ALL name
	DST	dstname
	OP	opid PPT ''
	REQ	requestnumber ALL
RESTORE	DELETE	
	TIMER	
RUNCMD	SP	service-point-name
	APPL	application-name
	CLISTVAR	YES NO
RUNDIAG	ID	name
	PORT	portnumber
	LINE	name
	TEST	testid
START	TASK	taskname DSILOG DSIPRT DSITRACE
STOP	FORCE	taskname hclname nntname terminalname DSILOG DSITRACE
	TASK	taskname
SWITCH	taskname	
	DSILOG	
	DSITRACE	

Command	Keyword	Value
THRESH	ID	name
	STATION	name
	QUERY	thrvalue
	PT	thrvalue
	TDT	thrvalue
	TET	thrvalue
	RDT	thrvalue
	RET	thrvalue
TRANSMMSG	MEMBER	member-name

Note: Most defaults are not scope checked. If a command has a default for a keyword such as:

DISPCMD ALL

where ALL is the default if only the command DISPCMD is entered, then ALL is only scope checked when explicitly stated and not scope checked when it is the default.

Hardware Monitor

To scope check a hardware monitor command, treat the command as a keyword to the NPDA command. The following table shows the hardware monitor commands that can be scope checked as keywords to NPDA.

Command	Keyword	Value
NPDA	PURGE/PRG	
	PRGATT	
	REPORTS	
	SRATIO/SR	
	SRFILTER/SRF	
	SWRAP/SW	
	DEL (see note 3)	

Notes:

1. For the hardware monitor, CMD==> is the same as the NPDA command. Therefore, both NPDA and CMD==> must be scope checked in order for scope checking to fully function from all components.
2. When scope checking a keyword that can be abbreviated, both the keyword and its abbreviation must be scope checked.
3. DEL is a panel option that may be scope checked as a keyword.

Session Monitor

To scope check a session monitor command, treat the command as a keyword to the NLDM command. The following table shows the session monitor commands that can be scope checked as keywords to NLDM.

Command	Keyword	Value
NLDM	COLLECT	
	DISABLE	
	DISKEEP	-
	ENABLE	-
	FORCE	
	KEEP	
	LINEMAP	
	LIST	
	PURGE	
	RECORD	
	RELOAD	
	RTREND	
	RTSUM	
	SENSE	
SESS		
TRACE		

Status Monitor

Command	Keyword	Value
MONIT	START	
	STOP	

Glossary, Bibliography, and Index

Glossary	147
Bibliography	169
NetView Publications	169
NetView/PC Publications	170
Other Network Program Products Publications	170
VTAM Publications	170
NCP, SSP, and EP Publications	170
Index	173

Glossary

This glossary defines important NCP, NetView, NetView/PC, SSP, and VTAM abbreviations and terms. It includes information from the *IBM Dictionary of Computing*, SC20-1699. Definitions from the *American National Dictionary for Information Processing* are identified by an asterisk (*). Definitions from draft proposals and working papers under development by the International Standards Organization, Technical Committee 97, Subcommittee 1 are identified by the symbol **(TC97)**. Definitions from the *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the Consultative Committee on International Telegraph and Telephone of the International Telecommunication Union, Geneva, 1980 are preceded by the symbol **(CCITT/ITU)**. Definitions from published sections of the *ISO Vocabulary of Data Processing*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1 and from published sections of the *ISO Vocabulary of Office Machines*, developed by subcommittees of ISO Technical Committee 95, are preceded by the symbol **(ISO)**.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

Reference Words Used in the Entries

The following reference words are used in this glossary:

Deprecated term for. Indicates that the term should not be used. It refers to a preferred term, which is defined.

Synonymous with. Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

Synonym for. Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

Contrast with. Refers to a term that has an opposed or substantively different meaning.

See. Refers to multiple-word terms that have the same last word.

See also. Refers to related terms that have similar (but not synonymous) meanings.

ACB. (1) In VTAM, access method control block.
(2) In NCP, adapter control block.

ACB name. (1) The name of an ACB macroinstruction.
(2) A name specified in the ACBNAME parameter of a VTAM APPL statement. Contrast with *network name*.

accept. For a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macroinstruction, or a VTAM operator issues a command. See also *acquire (1)*.

access method. A technique for moving data between main storage and input/output devices.

access method control block (ACB). A control block that links an application program to VSAM or VTAM.

accounting exit routine. In VTAM, an optional installation exit routine that collects statistics about session initiation and termination.

ACF/NCP. Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

ACF/SSP. Advanced Communications Function for the System Support Programs. Synonym for *SSP*.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

acquire. (1) For a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*.
(2) To take over resources that were formerly controlled by an access method in another domain, or to resume control of resources that were controlled by this domain but released. Contrast with *release*. See also *resource takeover*.

activate. To make a resource of a node ready to perform the functions for which it was designed. Contrast with *deactivate*.

active. (1) The state a resource is in when it has been activated and is operational. Contrast with *inactive*, *pending*, and *inoperative*. (2) Pertaining to a major or minor node that has been activated by VTAM. Most resources are activated as part of VTAM start processing or as the result of a VARY ACT command.

adapter. Hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device.

adapter control block (ACB). In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

adaptive session pacing. Synonym for *adaptive session-level pacing*.

adaptive session-level pacing. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission to adapt dynamically to variations in availability and demand of buffers on a session by session basis. Session pacing occurs within independent stages along the session path according to local congestion at the intermediate nodes. Synonymous with *adaptive session pacing*. See *pacing*, *session-level pacing*, and *virtual route pacing*.

alert. (1) In SNA, a record sent to a system problem management focal point to communicate the existence of an alert condition. (2) In the NetView program, a high priority event that warrants immediate attention. This data base record is generated for certain event types that are defined by user-constructed filters.

alias name. A name defined in a host used to represent a logical unit name, logon mode table name, or class-of-service name in another network. This name is defined to a name translation program when the alias name does not match the real name. The alias name translation program is used to associate the real and alias names.

allocate. A logical unit (LU) 6.2 application program interface (API) verb used to assign a session to a conversation for the conversation's use. Contrast with *deallocate*.

API. Application program interface.

application program. (1) A program written for or by a user that applies to the user's work. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program identification. The symbolic name by which an application program is identified to VTAM. It is specified in the APPLID parameter of the ACB macroinstruction.

application program interface (API). (1) The formally defined programming language interface between an IBM system control program or licensed program and its user. (2) The interface through which an application program interacts with an access method. In VTAM, it is the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

attaching device. Any device that is physically connected to a network and can communicate over the network.

authorization exit routine. In VTAM, an optional installation exit routine that approves or disapproves requests for session initiation.

authorized receiver. In the NetView program, an authorized operator who receives all the unsolicited and authorized-receiver messages not assigned to a specific operator.

automatic logon. (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session will be between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. See also *controlling application program* and *controlling logical unit*. (2) In VM, a process by which a virtual machine is initiated by other than the user of that virtual machine. For example, the primary VM operator's virtual machine is activated automatically during VM initialization.

automatic reactivation. In the NetView program, the activation of a node from the inactive state without any action by the network operator.

available. In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

begin bracket. In SNA, the value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *end bracket*. See also *bracket*.

bidder. In SNA, the LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session to begin a bracket. Contrast with *first speaker*. See also *bracket protocol* and *contention*.

binary synchronous communication (BSC). (1) Communication using binary synchronous line discipline. (2) A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

bind. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

BIU segment. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either a request/response header (RH) followed by all or a portion of a request/response unit (RU), or only a portion of an RU.

blocking of PIUs. In SNA, an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

boundary function. (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *boundary node*, *network addressable unit (NAU)*, *peripheral path control*, *subarea node*, and *subarea path control*.

boundary node. (1) A subarea node with boundary function. See *subarea node* (including illustration). See also *boundary function*. (2) The programming component that performs FID2 (format identification type 2) conversion, channel data link control, pacing, and channel or device error recovery procedures for a locally attached station. These functions are similar to those performed by a network control program for an NCP-attached station.

bracket. In SNA, one or more chains of request units (RUs) and their responses that are exchanged between the two LU-LU half-sessions and that represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are data base inquiries/replies, update transactions, and remote job entry output sequences to work stations. See also *begin bracket* and *end bracket*.

bracket protocol. In SNA, a data flow control protocol in which exchanges between the two LU-LU half-sessions are achieved through the use of brackets, with one LU designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder* and *first speaker*.

browse. A way of looking at a file that does not allow you to change it.

BSC. Binary synchronous communication.

buffer. A portion of storage for temporarily holding input or output data.

call. (1) * (ISO) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) To transfer control to a procedure, program, routine, or subroutine. (3) The actions necessary to make a connection between two stations. (4) To attempt to contact a user, regardless of whether the attempt is successful.

CALLOUT. The logical channel type on which the data terminal equipment (DTE) can send a call, but cannot receive one.

calling. * (ISO) The process of transmitting selection signals in order to establish a connection between data stations.

CCS. Console communication services.

CDRM. Cross-domain resource manager.

CDRSC. Cross-domain resource.

channel. * A path along which signals can be sent, for example, data channel, output channel. See *data channel* and *input/output channel*. See also *link*.

channel-attached. (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

channel link. A System/370 I/O channel to control unit interface that has an SNA network address. A channel link can be either a subarea link or a peripheral link. See also *link* and *subarea link*.

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

CLIST. Command list.

CMS. Conversational Monitor System.

CNM. Communication network management.

code point. In the NetView/PC program and in the NetView program, a 1- or 2-byte hexadecimal value that indexes a text string stored at an alert receiver and is used by the alert receiver to create displays of alert information.

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in a FID4 transmission header. See also *VTAM operator command*.

command facility. The component of the NetView program that is a base for command processors that can monitor, control, automate, and improve the operation of a network.

command list. A list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in NetView command list language.

command procedure. Either a command list or a command processor. See also *command list* and *command processor*.

command processor. (1) A program that performs an operation specified by a command. (2) In the NetView program, a user-written module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are invoked as commands.

communication line. Deprecated term for *telecommunication line* and *transmission line*.

communication management configuration host node. The type 5 host processor in a communication management configuration that does all network-control functions in the network except for the control of devices channel-attached to data hosts. Synonymous with *communication management host*. Contrast with *data host node*.

communication management host. Synonym for *communication management configuration host node*. Contrast with *data host*.

communication network management (CNM). The process of designing, installing, operating, and managing the distribution of information and controls among end users of communication systems.

communication network management (CNM) application program. A VTAM application program that issues and receives formatted management services request units for physical units. For example, the NetView program.

communication network management (CNM) interface. The interface that the access method provides to an application program for handling data and commands associated with communication system management. CNM data and commands are handled across this interface.

communication network management (CNM) processor. A program that manages one of the functions of a communications system. A CNM processor is executed under control of the NetView program.

component. A command that (a) controls the terminal's screen (using the DSIPSS macro (TYPE=ASYPANEL) or the VIEW command), (b) allows the operator to enter NetView commands, and (c) can resume when such commands are complete.

composite end node (CEN). A group of nodes made up of a single type 5 node and its subordinate type 4 nodes

that together support type 2.1 protocols. To a type 2.1 node, a CEN appears as one end node. For example, NCP and VTAM act as a composite end node.

configuration. (1) (TC97) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

configuration services. In SNA, one of the types of network services in the control point (CP) and in the physical unit (PU); configuration services activate, deactivate, and maintain the status of physical units, links, and link stations. Configuration services also shut down and restart network elements and modify path control routing tables and address-translation tables. See also *maintenance services*, *management services*, *network services*, and *session services*.

connected. In VTAM, pertaining to a physical unit (PU) or logical unit (LU) that has an active physical path to the host processor containing the system services control point (SSCP) that controls the PU or LU.

connection. Synonym for *physical connection*.

console communications services (CCS). The SNA facility that acts as an interface between the control program and the VSCS component of VTAM for VM.

contention. A situation in which two logical units (LUs) that are connected by an LU 6.2 session both attempt to allocate the session for a conversation at the same time. The control operator assigns "winner" and "loser" status to the LUs so that processing may continue on an orderly basis. The contention loser requests permission from the contention winner to allocate a conversation on the session, and the contention winner either grants or rejects the request. See also *bidder*.

control point (CP). (1) A system services control point (SSCP) that provides hierarchical control of a group of nodes in a network. (2) A control point (CP) local to a specific node that provides control of that node, either in the absence of SSCP control (for type 2.1 nodes engaged in peer to peer communication) or to supplement SSCP control.

control program (CP). The VM operating system that manages the real processor's resources and is responsible for simulating System/370s for individual users.

controlling application program. In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in

session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit*.

controlling logical unit. In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program*.

control statement. A statement in a command list that controls the processing sequence of the command list or allows the command list to send messages to the operator and receive input from the operator.

conversation. In SNA, a logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

Conversational Monitor System (CMS). A VM application program for general interactive time sharing, problem solving, and program development.

COS. Class of service.

coupler. A hardware device that connects a modem to a public phone system in much the same way that a telephone does.

CP. (1) Control program. (2) Control point.

cross-domain. In SNA, pertaining to control of resources involving more than one domain.

cross-domain resource (CDRSC). A resource owned by a cross-domain resource manager (CDRM) in another domain but known by the CDRM in this domain by network name and associated CDRM.

cross-domain resource manager (CDRM). In VTAM, the function in the system services control point (SSCP) that controls initiation and termination of cross-domain sessions.

cross-network. In SNA, pertaining to control or resources involving more than one SNA network.

cross-network session. An LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

DASD. Direct access storage device.

data channel. Synonym for *input/output channel*. See *channel*.

data circuit-terminating equipment (DCE). (TC97) The equipment installed at the user's premises that provides all functions required to establish, maintain, and

terminate a connection, and the signal conversion and coding between the data terminal equipment (DTE) and the line. The DCE may be separate equipment or an integral part of other equipment.

data host. Synonym for *data host node*. Contrast with *communication management configuration host*.

data host node. In a communication management configuration, a type 5 host node that is dedicated to processing applications and does not control network resources, except for its channel-attached or communication adapter-attached devices. Synonymous with *data host*. Contrast with *communication management configuration host node*.

data link. In SNA, synonym for *link*.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a transmission medium connecting two nodes and perform error control for the link connection. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

data services command processor (DSCP). A component that structures a request for recording and retrieving data in the application program's data base and for soliciting data from a device in the network.

data services task (DST). The NetView subtask that gathers, records, and manages data in a VSAM file and/or a network device that contains network management information.

data terminal equipment (DTE). (TC97) That part of a data station that serves as a data source, data link, or both, and provides for the data communication control function according to protocols.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data types. In the NetView program, a concept to describe the organization of panels. Data types are defined as alerts, events, and statistics. Data types are combined with resource types and display types to describe NetView's display organization. See also *display types* and *resource types*.

DBCS. Double-byte character set.

DCE. Data circuit-terminating equipment.

deactivate. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

deallocate. A logical unit (LU) 6.2 application program interface (API) verb that terminates a conversation, thereby freeing the session for a future conversation. Contrast with *allocate*.

definite response (DR). In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response* and *no response*.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See Figure 1, Figure 2, and Figure 3. See also *macro-instruction*.

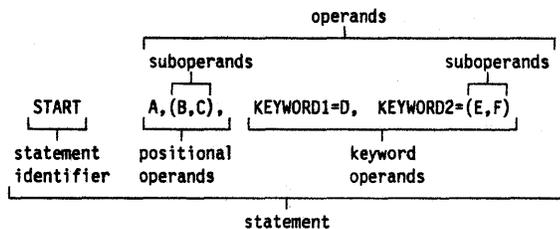


Figure 1. Example of a Language Statement

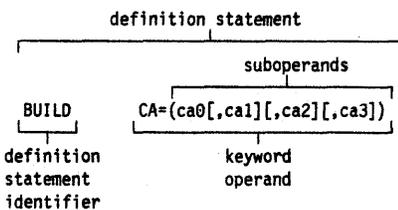


Figure 2. NCP Examples

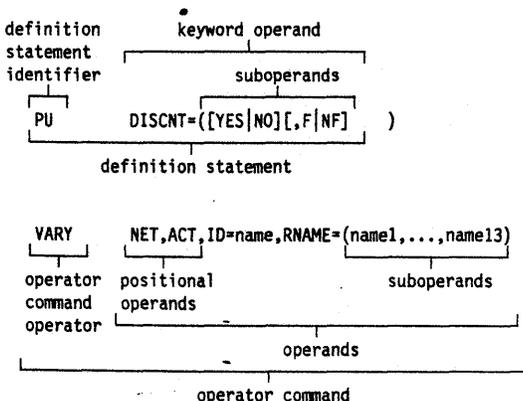


Figure 3. VTAM Examples

device. An input/output unit such as a terminal, display, or printer. See *attaching device*.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data. For example, a disk.

directory. In VM, a control program (CP) disk that defines each virtual machine's normal configuration.

disabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited*.

display. (1) To present information for viewing, usually on a terminal screen or a hard-copy device. (2) A device or medium on which information is presented, such as a terminal screen. (3) Deprecated term for *panel*.

display levels. Synonym for *display types*.

display types. In the NetView program, a concept to describe the organization of panels. Display types are defined as total, most recent, user action, and detail. Display types are combined with resource types and data types to describe NetView's panel organization. See *data types* and *resource types*. Synonymous with *display levels*.

domain. (1) An access method, its application programs, communication controllers, connecting lines, modems, and attached terminals. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See *system services control point domain* and *type 2.1 node control point domain*. See also *single-domain network* and *multiple-domain network*.

domain operator. In a multiple-domain network, the person or program that controls the operation of the resources controlled by one system services control point. Contrast with *network operator* (2).

double-byte character set (DBCS). A character set, such as Japanese, in which each character is represented by a two-byte code.

downstream. In the direction of data flow from the host to the end user. Contrast with *upstream*.

drop. In the IBM Token-Ring Network, a cable that leads from a faceplate to the to the distribution panel in a wiring closet. When the IBM Cabling System is used with the IBM Token-Ring Network, a drop may form part of a lobe.

DSCP. Data services command processor.

DST. Data services task.

DTE. Data terminal equipment.

dump. (1) Computer printout of storage. (2) To write the contents of all or part of storage to an external medium as a safeguard against errors or in connection with debugging. (3) (ISO) Data that have been dumped.

duplex. * In data communication, pertaining to a simultaneous two-way independent transmission in both directions. Synonymous with *full duplex*. Contrast with *half duplex*.

EBCDIC. * Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

echo. The return of characters to the originating SS device to verify that a message was sent correctly.

ECL. Electronic cabling link.

element. (1) A field in the network address. (2) The particular resource within a subarea identified by the element address. See also *subarea*.

Emulation Program (EP). An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program*.

enabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is now ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *disabled* and *inhibited*.

end bracket. In SNA, the value (binary 1) of the end bracket indicator in the request header (RH) of the first request of the last chain of a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*. See also *bracket*.

end node. A type 2.1 node that does not provide any intermediate routing or session services to any other node. For example, APPC/PC is an end node. See *composite end node*, *node*, and *type 2.1 node*.

entry point. An SNA node that provides distributed network management support. It may be a type 2, type 2.1, type 4, or type 5 node. It sends SNA-formatted network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal point initiated commands to manage and control its resources.

EP. Emulation Program.

ER. (1) Explicit route. (2) Exception response.

error-to-traffic (E/T). The number of temporary errors compared to the traffic associated with a resource.

E/T. Error-to-traffic.

event. (1) In the NetView program, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as an input/output operation.

exception response (ER). In SNA, a value in the form-of-response-requested field of a request header (RH). An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response* and *no response*. See also *negative response*.

EXEC. In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

exit routine. Any of several types of special-purpose user-written routines. See *accounting exit routine*, *authorization exit routine*, *logon-interpret routine*, *virtual route selection exit routine*, *EXLST exit routine*, and *RPL exit routine*.

EXLST exit routine. In VTAM, a routine whose address has been placed in an exit list (EXLST) control block. The addresses are placed there with the EXLST macro-instruction, and the routines are named according to their corresponding operand; hence DFASY exit routine, TPEND exit routine, RELREQ exit routine, and so forth. All exit list routines are coded by the VTAM application programmer. Contrast with *RPL exit routine*.

explicit route (ER). In SNA, the path control network elements, including a specific set of one or more transmission groups, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*. See also *path* and *route extension*.

FDX. Full duplex.

feature. A particular part of an IBM product that a customer can order separately.

filter. In the NetView program, a function that limits the data that is to be recorded on the data base and displayed at the terminal. See *recording filter* and *viewing filter*.

first speaker. In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both half-sessions attempt to begin a bracket simultaneously. Contrast with *bidder*. See also *bracket protocol*.

flow control. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units, with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *adaptive session-level pacing*, *pacing*, *session-level pacing*, and *virtual route pacing*.

focal point. An entry point that provides centralized management and control for other entry points for one or more network management categories.

frame. (1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

full duplex (FDX). Synonym for *duplex*.

gateway. (1) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP. (2) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols.

generation. The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

generic alert. Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as the-NetView program.

generic unbind. Synonym for *session deactivation request*.

group. In the NetView/PC program, to identify a set of application programs that are to run concurrently.

half-duplex. * In data communication, pertaining to an alternate, one way at a time, independent transmission. Contrast with *duplex*.

half-session. In SNA, a component that provides function management data (FMD) services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also *primary half-session* and *secondary half-session*.

hard copy. A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, summaries, or network logs.

hard-copy task (HCT). The NetView subtask that controls the passage of data between the NetView program and the hard-copy device.

hardware monitor. The component of the NetView program that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques.

help panel. An online display that tells you how to use a command or another aspect of a product. See *task panel*.

High Performance Option (HPO). A licensed program that is an extension of VM/SP. It provides performance and operation enhancements for large system environments.

host node. A node providing an application program interface (API) and a common application interface. See *boundary node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

HPO. High Performance Option.

immediate command. In the NetView program, a command (such as GO, CANCEL, or RESET) that can be executed while a regular command is being processed.

inactive. Describes the state of a resource that has not been activated or for which the VARY INACT command has been issued. Contrast with *active*. See also *inoperative*.

information (I) format. A format used for information transfer.

inhibited. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a

primary logical unit (PLU) or as a secondary logical unit (SLU). See also *enabled* and *disabled*.

Inoperative. The condition of a resource that has been active, but is not. The resource may have failed, received an INOP request, or is suspended while a reactivate command is being processed. See also *inactive*.

Input/output channel. (1) (ISO) In a data processing system, a functional unit that handles the transfer of data between internal and peripheral equipment. (2) In a computing system, a functional unit, controlled by a processor, that handles the transfer of data between processor storage and local peripheral devices. Synonymous with *data channel*. See *channel*. See also *link*.

Interconnected networks. SNA networks connected by gateways.

Interface. * A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

Item. In CCP, any of the components, such as communication controllers, lines, cluster controllers, and terminals, that comprise an IBM 3710 Network Controller configuration.

Kanji. An ideographic character set used in Japanese. See also *double-byte character set*.

Katakana. A phonetic character set used in Japanese. It belongs to a single-byte character set.

keyword. (1) (TC97) A lexical unit that, in certain contexts, characterizes some language construction. (2) * One of the predefined words of an artificial language. (3) One of the significant and informative words in a title or document that describes the content of that document. (4) A name or symbol that identifies a parameter. (5) A part of a command operand that consists of a specific character string (such as *DSNAME=*). See also *definition statement* and *keyword operand*. Contrast with *positional operand*.

keyword operand. An operand that consists of a keyword followed by one or more values (such as *DSNAME=HELLO*). See also *definition statement*. Contrast with *positional operand*.

keyword parameter. A parameter that consists of a keyword followed by one or more values.

LAN. An industry-wide acronym for local area network.

LCC. Link connection component.

LCS. Link connection subsystem.

line. See *communication line*.

link. In SNA, the combination of the link connection and the link stations joining network nodes; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with *data link*. See Figure 4 on page 156.

link-attached. Pertaining to devices that are physically connected by a telecommunication line. Contrast with *channel-attached*. Synonymous with *remote*.

link connection. In SNA, the physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit terminating equipment (DCE).

link connection component (LCC). Components of the link that perform functions for the physical layer of the link.

link connection segment. A portion of the configuration that is located between two resources listed consecutively in the service point command service (SPCS) query link configuration request list.

link connection subsystem (LCS). The sequence of link connection components (LCCs) that belong to a link connection and are managed by one LCSM.

Link Problem Determination Aid (LPDA). A series of testing procedures initiated by the NetView program that provide modem status, attached device status, and the overall quality of a communications link.

link station. (1) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. (2) In VTAM, a named resource within a subarea node that represents another subarea node that is attached by a subarea link. In the resource hierarchy, the link station is subordinate to the subarea link.

load module. (ISO) A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor.

local. Pertaining to a device that is attached to a controlling unit by cables, rather than by a telecommunication line. Synonymous with *channel-attached*.

local address. In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

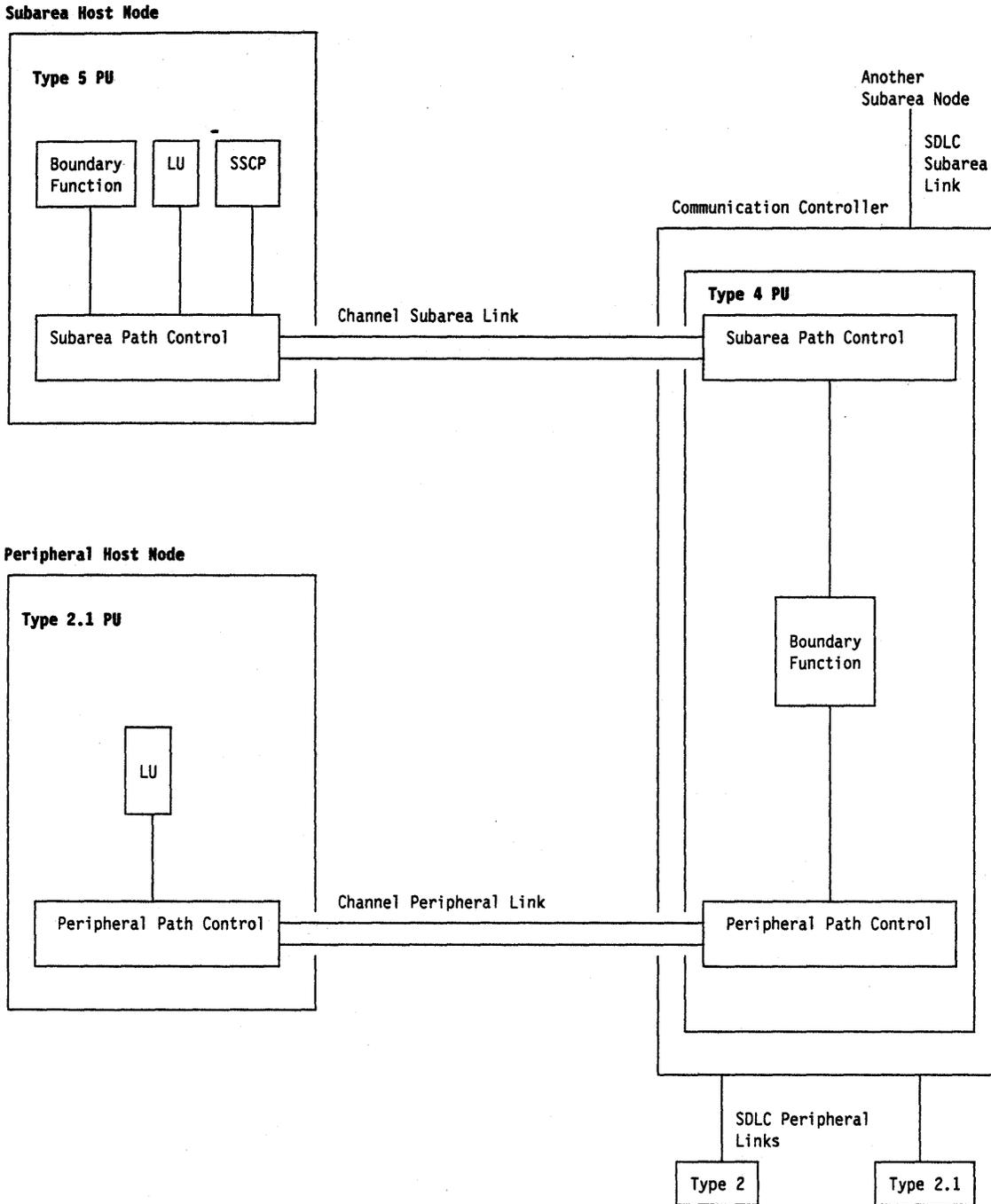


Figure 4. Links and Path Controls

local area network (LAN). (1) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately sized geographic area such as a single office building, warehouse, or campus, and which do not generally extend across public rights-of-way. Contrast with *wide area network*.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other LUs. See also *network addressable unit (NAU)*, *peripheral LU*, *phys-*

ical unit (PU), system services control point (SSCP), primary logical unit (PLU), and secondary logical unit (SLU).

logical unit (LU) services. In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also *physical unit (PU) services*.

logical unit (LU) 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (1) a peer relationship between session partners, (2) efficient utilization of a session for multiple transactions, (3) comprehensive end-to-end error processing, and (4) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

logoff. In VTAM, an unformatted session termination request.

log on. To initiate a session.

logon. In VTAM, an unformatted session initiation request for a session between two logical units. See *automatic logon* and *simulated logon*. See also *session-initiation request*.

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

logon-interpret routine. In VTAM, an installation exit routine, associated with an interpret table entry, that translates logon information. It may also verify the logon.

LPDA. Link Problem Determination Aid.

LU. Logical unit.

LU type. In SNA, the classification of an LU-LU session in terms of the specific subset of SNA protocols and options supported by the logical units (LUs) for that session, namely:

The mandatory and optional values allowed in the session activation request.

The usage of data stream controls, function management headers (FMHs), request unit (RU) parameters, and sense codes.

Presentation services protocols such as those associated with FMH usage.

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU-LU session type. A deprecated term for *LU type*.

LU 6.2. Logical unit 6.2.

macroinstruction. (1) An instruction that when executed causes the execution of a predefined sequence of instructions in the same source language. (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. See also *definition statement*.

maintenance services. In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). Maintenance services provide facilities for testing links and nodes and for collecting and recording error information. See also *configuration services, management services, network services, and session services*.

major node. In VTAM, a set of resources that can be activated and deactivated as a group. See *node* and *minor node*.

management services. In SNA, one of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management and change management. See also *configuration services, maintenance services, network services, and session services*.

Medium Access Control (MAC). The sublayer of DLC that supports medium-dependent functions and uses the services of the physical layer to provide services to Logical Link Control (LLC). The MAC sublayer includes the medium access port.

medium access control (MAC) procedure. (TC97) In a local area network, the part of the protocol that governs access to the transmission medium independently of the physical characteristics of the medium, but takes into account the topological aspects of the network, in order to enable the exchange of data between data stations.

message. (1) (TC97) A group of characters and control bit sequences transferred as an entity. (2) In VTAM, the amount of function management data (FMD) transferred to VTAM by the application program with one SEND request.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

minor node. In VTAM, a uniquely-defined resource within a major node. See *node* and *major node*.

modem. A device that modulates and demodulates signals transmitted over data communication facilities. The term is a contraction for modulator-demodulator.

mode name. A symbolic name for a set of session characteristics. For LU 6.2, a mode name and a partner LU name together define a group of parallel sessions having the same characteristics.

module. * A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

monitor. In the IBM Token-Ring Network, the function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

multiple-domain network. In SNA, a network with more than one system services control point (SSCP). Contrast with *single-domain network*.

Multiple Virtual Storage (MVS). An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

Multiple Virtual Storage for Extended Architecture (MVS/XA). An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for Extended Architecture. Extended architecture allows 31-bit storage addressing. MVS/XA is a software operating system controlling the execution of programs.

MVS. Multiple Virtual Storage operating system.

MVS/XA. Multiple Virtual Storage for Extended Architecture operating system.

NAM. network asset management facility

name translation. In SNA network interconnection, converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network. This function can be provided through the NetView program and invoked by a gateway system services control point (SSCP) when necessary. See also *alias name*.

NAU. Network addressable unit.

NCCF. Network Communications Control Facility.

NCP. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

NCP major node. In VTAM, a set of minor nodes representing resources, such as lines and peripheral nodes, controlled by a network control program. See *major node*.

negative response (NR). In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*. See *exception response*.

NetView. A system 370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

NetView command list language. An interpretive language unique to the NetView program that is used to write command lists.

NetView-NetView task (NNT). The task under which a cross-domain NetView operator session runs. See *operator station task*.

NetView/PC. A PC-based IBM licensed program through which application programs can be used to monitor, manage, and diagnose problems in IBM Token-Ring networks, non-SNA communication devices, and voice networks.

network. (1) (TC97) An interconnected group of nodes. (2) In data processing, a user application network. See *path control network*, *public network*, *SNA network*, *subarea network*, *type 2.1 network*, and *user-application network*.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address*. See also *network name*.

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*, and *path control network*.

Network Communications Control Facility (NCCF). An IBM licensed program that is a base for command processors that can monitor, control, automate, and improve the operations of a network. Its function is included and enhanced in NetView's command facility.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and inter-connected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network control program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

network identifier (network ID). The network name defined to NCPs and hosts to indicate the name of the network in which they reside. It is unique across all communicating SNA networks. communication among domains.

network log. A file that contains all messages processed by the NetView program.

Network Logical Data Manager (NLDM). An IBM licensed program that collects and correlates session-related data and provides online access to this information. It runs as an NCCF communication network management (CNM) application program. Its function is included and enhanced in NetView's session monitor.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network name. (1) In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link, or a link station. See also *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program is its network name and it must be unique across domains. Contrast with *ACB name*. See *uninterpreted name*.

network operator. (1) A person or program responsible for controlling the operation of all or part of a network. (2) The person or program that controls all the domains in a multiple-domain network. Contrast with *domain operator*.

Network Problem Determination Application (NPDA). An IBM licensed program that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. It runs as an NCCF communication network management (CNM) application

program. Its function is included and enhanced in NetView's hardware monitor.

network services (NS). In SNA, the services within network addressable units (NAUs) that control network operation through SSCP-SSCP, SSCP-PU, and SSCP-LU sessions. See *configuration services, maintenance services, management services, and session services*.

NLDM. Network Logical Data Manager.

NMVT. Network management vector transport.

NNT. NetView-NetView task.

node. (1) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. See *boundary node, host node, peripheral node, and subarea node* (including illustration). (2) In VTAM, a point in a network defined by a symbolic name. See *major node and minor node*.

node name. In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes. See also *type 2.1 node*.

nonswitched connection. A connection that does not have to be established by dialing. Contrast with *switched connection*.

no response. In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

notify. A network services request that is sent by an SSCP to a logical unit (LU) to inform the LU of the status of a procedure requested by the LU.

NPDA. Network Problem Determination Application.

online. Stored in a computer and accessible from a terminal.

open. (1) In the IBM Token-Ring Network, to make an adapter ready for use. (2) A break in an electrical circuit.

operand. (1) (ISO) An entity on which an operation is performed. (2) * That which is operated upon. An

operand is usually identified by an address part of an instruction. (3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor. (4) An expression to whose value an operator is applied. See also *definition statement*, *keyword*, *keyword parameter*, and *parameter*.

operator. (1) In a language statement, the lexical entity that indicates the action to be performed on operands. (2) A person who operates a machine. See *network operator*. See also *definition statement*.

operator profile. In the NetView program, the resources and activities a network operator has control over. The statements defining these resources and activities are stored in a file that is activated when the operator logs on.

operator station task (OST). The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

OST. Operator station task.

padding. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See *session-level pacing*, *send pacing*, and *virtual route (VR) pacing*. See also *flow control*.

pacing response. In SNA, an indicator that signifies a receiving component's readiness to accept another pacing group; the indicator is carried in a response header (RH) for session-level pacing, and in a transmission header (TH) for virtual route pacing.

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also *scroll*. (3) (ISO) In a virtual storage system, a fixed-length block that has a virtual address and that can be transferred between real storage and auxiliary storage. (4) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. See also *help panel* and *task panel*. Contrast with *screen*. (2) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

parameter. (1) (ISO) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user

or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. See also *keyword*, *keyword parameter*, and *operand*.

parallel sessions. In SNA, two or more concurrently active sessions between the same two logical units (LUs) using different pairs of network addresses. Each session can have independent session parameters.

path. (1) In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*. (2) In VTAM when defining a switched major node, a potential dial-out port that can be used to reach that node. (3) In the NetView/PC program, a complete line in a configuration that contains all of the resources in the service point command service (SPCS) query link configuration request list.

path control (PC). The function that routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units (BTUs) and one or more PIUs with data link control. Path control differs for peripheral nodes, which use local addresses for routing, and subarea nodes, which use network addresses for routing. See *peripheral path control* and *subarea path control*. See also *link*, *peripheral node*, and *subarea node*.

path control (PC) layer. In SNA, the layer that manages the sharing of link resources of the SNA network and routes basic information units (BIUs) through it. See also *BIU segment*, *blocking of PIUs*, *data link control layer*, and *transmission control layer*.

path control (PC) network. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network* and *user application network*. See also *boundary function*.

path information unit (PIU). In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header*.

PC. (1) Path control. (2) Personal Computer. Its full name is the IBM Personal Computer.

performance class. In the NetView program, a description of an objective or commitment of performance. It consists of a performance class name, boundary definitions, response time definition, response time ranges, and response time percentage objectives. Sessions may be assigned performance classes.

peripheral host node. A node that provides an application program interface (API) for running application programs but does not provide SSCP functions and is not aware of the network configuration. The peripheral host node does not provide subarea node services. It has boundary function provided by its adjacent subarea. See *boundary node*, *host node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral LU. In SNA, a logical unit representing a peripheral node.

peripheral node. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node is a physical unit (PU) type 1, 2.0, or 2.1 node connected to a subarea node with boundary function within a subarea. See *boundary node*, *host node*, *node*, *peripheral host node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral path control. The function in a peripheral node that routes message units between units with local addresses and provides the paths between them. See *path control* and *subarea path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

peripheral PU. In SNA, a physical unit representing a peripheral node.

Personal Computer (PC). The IBM Personal Computer line of products including the 5150 and subsequent models.

physical connection. In VTAM, a point-to-point connection or multipoint connection. Synonymous with *connection*.

physical unit (PU). In SNA, a type of network addressable unit (NAU). A physical unit (PU) manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. See also *peripheral PU* and *subarea PU*.

physical unit (PU) services. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also *logical unit (LU) services*.

PIU. Path information unit.

PLU. Primary logical unit.

POI. Programmed operator interface.

positional operand. An operand in a language statement that has a fixed position. See also *definition statement*. Contrast with *keyword operand*.

positive response. A response indicating that a request was received and processed. Contrast with *negative response*.

PPT. Primary POI task.

primary half-session. In SNA, the half-session that sends the session activation request. See also *primary logical unit*. Contrast with *secondary half-session*.

primary logical unit (PLU). In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU)*.

primary POI task (PPT). The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when the NetView program is initialized and timer request commands scheduled to execute under the PPT.

problem determination. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environment failure such as a power loss, or a user error.

profile. In the Conversational Monitor System (CMS) or the group control system (GCS), the characteristics defined by a PROFILE EXEC file that executes automatically after the system is loaded into a virtual machine. See also *operator profile*.

programmed operator interface (POI). A VTAM function that allows programs to perform VTAM operator functions.

protocol. (1) (CCITT/ITU) A specification for the format and relative timing of information exchanged between communicating parties. (2) (TC97) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. See also *bracket protocol*. Synonymous with *line control discipline* and *line discipline*. See also *link protocol*.

PU. Physical unit.

PU type. Physical unit type.

public network. A network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet switched, and leased-circuit services to the public. Contrast with *user-application network*.

PU-PU flow. In SNA, the exchange between physical units (PUs) of network control requests and responses.

RACF. Resource Access Control Facility.

RDT. Resource definition table.

real name. The name by which a logical unit (LU), logon mode table, or class-of-service (COS) table is known within the SNA network in which it resides.

receive pacing. In SNA, the pacing of message units that the component is receiving. See also *send pacing*.

record. (1) (ISO) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (2) (TC97) A set of data treated as a unit. (3) A set of one or more related data items grouped for processing. (4) In VTAM, the unit of data transmission for record mode. A record represents whatever amount of data the transmitting node chooses to send.

recording filter. In the NetView program, the function that determines which events, statistics, and alerts are stored on a data base.

regular command. In the NetView program, any VTAM or NetView command that is not an immediate command and is processed by a regular command processor. Contrast with *immediate command*.

release. For VTAM, to relinquish control of resources (communication controllers or physical units). See also *resource takeover*. Contrast with *acquire (2)*.

remote. Concerning the peripheral parts of a network not centrally linked to the host processor and generally using telecommunication lines with public right-of-way.

remove. In the IBM Token-Ring Network, to take an attaching device off the ring.

request header (RH). In SNA, control information preceding a request unit (RU). See also *request/response header (RH)*.

request unit (RU). In SNA, a message unit that contains control information, end-user data, or both.

request/response header (RH). In SNA, control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

request/response unit (RU). In SNA, a generic term for a request unit or a response unit. See also *request unit (RU)* and *response unit*.

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

resource. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network.

Resource Access Control Facility (RACF). A licensed program that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected data sets.

resource definition table (RDT). In VTAM, a table that describes the characteristics of each node available to VTAM and associates each node with a network address. This is the main VTAM network configuration table.

resource level. In the NetView program, the hierarchical position of a device (and the software contained within it) in a data processing system. For example, a first-level resource would be the communication controller, and the second-level resource would be the line connected to it.

resource takeover. In VTAM, action initiated by a network operator to transfer control of resources from one domain to another. See also *acquire (2)* and *release*. See *takeover*.

resource types. In the NetView program, a concept to describe the organization of panels. Resource types are defined as central processing unit, channel, control unit, and I/O device for one category; and communication controller, adapter, link, cluster controller, and terminal for another category. Resource types are combined with data types and display types to describe display organization. See also *data types* and *display types*.

response. A reply represented in the control field of a response frame. It advises the primary or combined station of the action taken by the secondary or other combined station to one or more commands. See also *command*.

response header (RH). In SNA, a header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response*, *pacing response*, and *positive response*.

response time. (1) The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal. (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

response time monitor (RTM). A feature available with certain hardware devices to allow measurement of response times, which may be collected and displayed by the NetView program.

response unit (RU). In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

Restructured Extended Executor (REXX). An interpretive language used to write command lists.

REXX. Restructured Extended Executor.

RH. Request/response header.

ring. A network configuration where a series of attaching devices are connected by unidirectional transmission links to form a closed path.

route. See *explicit route* and *virtual route*.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path*, *explicit route (ER)* and *virtual route (VR)*.

routing. The assignment of the path by which a message will reach its destination.

RPL exit routine. In VTAM, an application program exit routine whose address has been placed in the EXIT field of a request parameter list (RPL). VTAM invokes the routine to indicate that an asynchronous request has been completed. See *EXLST exit routine*.

RTM. Response time monitor.

RU. Request/response unit.

SAW data. Synonym for *session awareness (SAW) data*.

scope of commands. In the NetView program, the facility that provides the ability to assign different responsibilities to various operators.

screen. An illuminated display surface; for example, the display surface of a CRT or plasma panel. Contrast with *panel*.

scroll. To move all or part of the display image vertically to display data that cannot be observed within a single display image. See also *page (2)*.

SDLC. Synchronous Data Link Control.

secondary half-session. In SNA, the half-session that receives the session-activation request. See also *secondary logical unit (SLU)*. Contrast with *primary half-session*.

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU)*.

secondary logical unit (SLU) key. A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

segment. (1) In the IBM Token-Ring Network, a section of cable between components or devices on the network. A segment may consist of a single patch cable, multiple patch cables connected together, or a combination of building cable and patch cables connected together. (2) See *link connection segment*.

send pacing. In SNA, pacing of message units that a component is sending. See also *receive pacing*.

sequence number. A number assigned to a particular frame or packet to control the transmission flow and receipt of data.

Service Level Reporter (SLR). A licensed program that generates management reports from data sets such as System Management Facility (SMF) files.

service point (SP). An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

service reminder (SR). In the NetView/PC program, a notification set by the operator that is displayed on a panel and logs a specified message.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session. See *half-session*, *LU-LU session*, *SSCP-LU session*, *SSCP-PU session*, and *SSCP-SSCP session*. See also *LU-LU session type* and *PU-PU flow*.

session activation request. In SNA, a request that activates a session between two network addressable units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Contrast with *session deactivation request*.

session awareness (SAW) data. Data collected by the NetView program about a session that includes the session type, the names of session partners, and information about the session activation status. It is collected for LU-LU, SSCP-LU, SSCP-PU, and SSCP-SSCP sessions and for non-SNA terminals not supported by NTO. It can be displayed in various forms, such as most recent sessions lists.

session data. Data about a session, collected by the NetView program, that consists of session awareness data, session trace data, and session response time data.

session deactivation request. In SNA, a request that deactivates a session between two network addressable units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

session-initiation request. In SNA, an Initiate or logon request from a logical unit (LU) to a control point (CP) that an LU-LU session be activated.

session-level pacing. In SNA, a flow control technique that permits a receiver to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing* and *virtual route pacing*.

session monitor. The component of the NetView program that collects and correlates session-related data and provides online access to this information.

session parameters. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units. See also *logon mode*.

session partner. In SNA, one of the two network addressable units (NAUs) having an active session.

session services. In SNA, one of the types of network services in the control point (CP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that the SSCP initiate or terminate sessions between logical units. See *configuration services*, *maintenance services*, and *management services*.

session trace. In the NetView program, the function that collects session trace data for sessions involving specified resource types or involving a specific resource.

session trace data. Data relating to sessions that is collected by the NetView program whenever a session trace is started and that consists of session activation parameters, VTAM path information unit (PIU) data, and NCP data.

shared. Pertaining to the availability of a resource to more than one use at the same time.

simulated logon. A session-initiation request generated when a VTAM application program issues a SIMLOGON macroinstruction. The request specifies a logical unit (LU) with which the application program wants a session in which the requesting application program will act as the primary logical unit (PLU).

single-domain network. In SNA, a network with one system services control point (SSCP). Contrast with *multiple-domain network*.

SLR. Service Level Reporter.

SLU. Secondary logical unit.

SMF. System management facility.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNA terminal. A terminal that supports Systems Network Architecture protocols.

SNBU. Switched network backup.

solicited message. A response from VTAM to a command entered by a program operator. Contrast with *unsolicited message*.

SP. Service point.

span. In the NetView program, a user-defined group of network resources within a single domain. Each major or minor node is defined as belonging to one or more spans. See also *span of control*.

span of control. The total network resources over which a particular network operator has control. All the network resources listed in spans associated through profile definition with a particular network operator are within that operator's span of control.

SR. Service reminder.

SS. Start-stop.

SSCP. System services control point.

SSCP ID. In SNA, a number that uniquely identifies a system services control point (SSCP). The SSCP ID is used in session activation requests sent to physical units (PUs) and other SSCPs.

SSCP-LU session. In SNA, a session between a system services control point (SSCP) and a logical unit (LU); the session enables the LU to request the SSCP to help initiate LU-LU sessions.

SSCP-PU session. In SNA, a session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

SSCP-SSCP session. In SNA, a session between the system services control point (SSCP) in one domain and the SSCP in another domain. An SSCP-SSCP session is used to initiate and terminate cross-domain LU-LU sessions.

SSP. System Support Programs (IBM licensed program). Its full name is Advanced Communications Function for System Support Programs. Synonymous with *ACF/SSP*.

ST. Session configuration screen abbreviation.

start option. In VTAM, a user-specified or IBM-supplied option that determines certain conditions that are to exist during the time a VTAM system is operating. Start options can be predefined or specified when VTAM is started.

statement. A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

station. (1) One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. (2) One or more

computers, terminals, or devices at a particular location.

status monitor. A component of the NetView program that collects and summarizes information on the status of resources defined in a VTAM domain.

subarea. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea host node. A host node that provides both subarea function and an application program interface (API) for running application programs. It provides system services control point (SSCP) functions, subarea node services, and is aware of the network configuration. See *boundary node, communication management configuration host node, data host node, host node, node, peripheral node, and subarea node*. See also *boundary function and node type*.

subarea link. In SNA, a link that connects two subarea nodes. See *channel link and link*.

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See *boundary node, host node, node, peripheral node, and subarea host node*. See also *boundary function and node type*.

subarea path control. The function in a subarea node that routes message units between network addressable units (NAUs) and provides the paths between them. See *path control and peripheral path control*. See also *boundary function, peripheral node, and subarea node*.

subarea PU. In SNA, a physical unit (PU) in a subarea node.

subsystem. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

subvector. A subcomponent of the MAC major vector.

supervisor. The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations.

supervisor call (SVC). A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system

from inappropriate user entry. All operating system requests must be handled by SVCs.

supervisor call (SVC) instruction. An instruction that interrupts the program being executed and passes control to the supervisor so that it can perform a specific service indicated by the instruction.

suppression character. In the NetView program, a user-defined character that is coded at the beginning of a command list statement or a command to prevent the statement or command from appearing on the operator's terminal screen or in the network log.

SVC. (1) Supervisor call. (2) Switched virtual circuit.

switched connection. (1) (TC97) A mode of operating a data link in which a circuit or channel is established to switching facilities, for example, in a public switched network. (2) A connection that is established by dialing. (3) Contrast with *nonswitched connection*.

switched major node. In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

switched network backup (SNBU). An optional facility that allows a user to specify, for certain types of PUs, a switched line to be used as an alternate path if the primary line becomes unavailable or unusable.

switched virtual circuit (SVC). An X.25 circuit that is dynamically established when needed. The X.25 equivalent of a switched line.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

system management facility (SMF). A standard feature of MVS that collects and records a variety of system and job-related information.

system services control point (SSCP). In SNA, a central location point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

system services control point (SSCP) domain. The system services control point and the physical units (PUs), logical units (LUs), links, link stations and all the resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

System Support Programs (SSP). An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

TAF. Terminal access facility.

takeover. The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem. See *resource takeover*.

task. A basic unit of work to be accomplished by a computer. The task is usually specified to a control program in a multiprogramming or multiprocessing environment.

task panel. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

TC. Transmission control.

telecommunication line. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

terminal. A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

terminal access facility (TAF). In the NetView program, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of such subsystems simultaneously.

TERMINATE. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

threshold. In the NetView program, refers to a percentage value set for a resource and compared to a calculated error-to-traffic ratio.

time-out. (1) (ISO) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

time sharing option (TSO). An optional configuration of the operating system that provides conversational time sharing from remote stations.

token. A sequence of bits passed from one device to another along the token ring. When the token has data appended to it, it becomes a frame.

token ring. A network with a ring topology that passes tokens from one attaching device to another. For example, the IBM Token-Ring Network.

transmission control (TC) layer. In SNA, the layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and decipheres end-user data. Transmission control has two components: the connection point manager and session control. See also *half-session*.

transmission group (TG). In SNA, a group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages. A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

transmission header (TH). In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

transmission line. Synonym for *telecommunication line*.

transmission priority. In SNA, a rank assigned to a path information unit (PIU) that determines its precedence for being selected by the transmission group control component of path control for forwarding to the next subarea node of the route used by the PIU.

TSO. Time sharing option.

type 2.1 node (T2.1 node). A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.0 nodes. Type 2.1 nodes can be directly attached to one another using peer-to-peer protocols. See *end node*, *node*, and *subarea node*. See also *node type*.

type 2.1 node (T2.1 node) control point domain. The CP, its logical units (LUs), links, link stations, and all resources that it activates and deactivates.

UNBIND. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

uninterpreted name. In SNA, a character string that a system services control point (SSCP) is able to convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

unsolicited message. A message, from VTAM to a program operator, that is unrelated to any command entered by the program operator. Contrast with *solicited message*.

upstream. In the direction of data flow from the end user to the host. Contrast with *downstream*.

user. Anyone who requires the services of a computing system.

user-application network. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. Contrast with *public network*.

user exit. A point in an IBM-supplied program at which a user routine may be given control.

user exit routine. A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language (HLL).

using node. (1) In NCP, the NCP in the host's domain that reports a link error condition. (2) For the command facility of the NetView program and for NCCF, the ID parameter of certain network control commands.

value. (1) (TC97) A specific occurrence of an attribute, for example, "blue" for the attribute "color." (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

variable. In the NetView program, a character string beginning with & that is coded in a command list and is assigned a value during execution of the command list.

verb. (1) In SNA, the general name for a transaction program's request for communication services. (2) In VTAM, a programming language element in the logical unit (LU) 6.2 application program interface (API) that causes an LU 6.2 function to be performed.

viewing filter. In the NetView program, the function that allows a user to select the data to be displayed on a terminal. All other stored data is blocked.

Virtual Machine (VM). A licensed program whose full name is the Virtual Machine/System Product (VM/SP). It is a software operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

virtual route (VR). In SNA, a logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER)*, *path*, and *route extension*.

virtual route (VR) pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing* and *session-level pacing*.

virtual route selection exit routine. In VTAM, an optional installation exit routine that modifies the list of virtual routes associated with a particular class of service before a route is selected for a requested LU-LU session.

virtual storage. (ISO) The notion of storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations.

Virtual Storage Access Method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Storage Extended (VSE). An IBM licensed program whose full name is the Virtual Storage

Extended/Advanced Function. It is a software operating system controlling the execution of programs.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

vital product data. Product identification information such as machine type, model number, and serial number for hardware products. For software products, vital product data can be version and release level.

VM. Virtual Machine operating system. Its full name is Virtual Machine/System Product. Synonymous with *VM/SP*.

VM/SP. Virtual Machine/System Product operating system. Synonym for *VM*.

VPD. vital product data

VR. Virtual route.

VSAM. Virtual Storage Access Method.

VSE. Virtual Storage Extended operating system. Synonymous with *VSE/AF*.

VSE/AF. Virtual Storage Extended/Advanced Function operating system. Synonym for *VSE*.

VTAM. Virtual Telecommunications Access Method (IBM licensed program). Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

VTAM definition. The process of defining the user application network to VTAM and modifying IBM-defined characteristics to suit the needs of the user.

VTAM operator command. A command used to monitor or control a VTAM domain. See also *definition statement*.

wide area network. A network that provides data communication capability in geographic areas larger than those serviced by local area networks. Wide area networks may extend across public rights-of-way. Contrast with *local area network*.

wrap. In general, to go from the maximum to the minimum in computer storage. For example, the continuation of an operation from the maximum value in storage to the first minimal value.

wrap count. In the NetView program, the number of events that can be retained on the data base for a specific resource.

Bibliography

NetView Publications

Learning About NetView: Operator Training (SK2T-0292) is an interactive PC-based operator training package that teaches SNA and basic network management concepts to new and inexperienced NetView operators. This training package uses graphics, animation and interactive NetView product simulations in a series of lessons to teach the basics of NetView operation.

NetView Installation and Administration Guide (SC31-6018) helps system programmers install and prepare the NetView program for operation. It is arranged in a simplified, step-by-step style and is meant to be used in conjunction with the sample network documented in *Network Program Products Samples*.

NetView Administration Reference (SC31-6014) is for system programmers and network operators who need a complete understanding of the NetView resource definition statements. This book lists each statement in alphabetical order giving its purpose and location.

NetView Tuning Guide (SC31-6079)¹ describes methods for controlling and improving the performance of the NetView Release 3 program. It is designed for system programmers who need to understand how NetView tuning values are determined and optimized.

NetView Customization Guide (SC31-6016) is designed for system programmers and others who want to customize the NetView program to reflect their network's needs or operating procedures. This book focuses on the different application programming interfaces that can be customized and explains how to modify NetView help panels and problem determination displays.

NetView Customization: Using PL/I and C (SC31-6037) describes the ways system programmers can tailor the NetView program to satisfy unique requirements or operating procedures. It discusses the uses and advantages of user-written programs (exit routines, command processors, and subtasks). It also provides instructions in designing, writing, and installing user-written programs in PL/I and C.

NetView Customization: Using Assembler (SC31-6078) describes the ways system programmers can tailor the NetView program to satisfy unique requirements or operating procedures. It discusses the uses and

advantages of user-written programs (exit routines, command processors, and subtasks). It also provides instructions in designing, writing, and installing user-written programs in Assembler.

NetView Customization: Writing Command Lists (SC31-6015) explains how to simplify network operator tasks by using command lists. It provides step-by-step instructions for writing simple and advanced command lists and for migrating from NCCF message automation to NetView message automation.

NetView Operation Primer (SC31-6020) provides a basic description of the network management task for new network operators. Topics include starting and stopping a network, controlling resources, monitoring a network, and gathering the necessary data to report a problem.

NetView Operation (SC31-6019) provides system programmers and experienced network operators a comprehensive explanation of network management using the NetView program. Topics include detailed command explanation and panel flows, as well as information on how the various components interact with each other.

NetView Command Summary (SX75-0026) is a reference card that provides network operators with the format of all the commands and the commonly used NetView command lists. The commands are listed in alphabetical order by component.

NetView Problem Determination and Diagnosis (LY43-0001) aids system programmers in identifying a NetView problem, classifying it, and describing it to an IBM Support Center.

NetView Problem Determination Supplement for Management Services Major Vectors 0001 and 0025 (LD21-0023) describes major vectors 0001 and 0025 for system programmers and network operators involved in problem determination or diagnosis. The supplement may be used for the generic alert option and other problem determination tasks.

NetView Resource Alerts Reference (SC31-6024) lists the messages sent by NetView-supported hardware and software resources. It helps system programmers analyze the messages into their component parts: action codes, event types, message text, and qualifiers. The book is a reference for those who need more information than online help provides.

¹ When available.

NetView Storage Estimates (SK2T-1988) is an interactive PC-based tool that helps the user estimate storage requirements for NetView. This tool can be used for planning, installation, and tuning purposes. It is intended for network planners, system programmers, and IBM service personnel.

Console Automation Using NetView: Planning (SC31-6058) describes an approach to automate the way a system handles messages and responses to alerts. It includes information you should know before beginning such automation, as well as sample plans and proposals you might find useful in promoting your automation concept. This book includes planning information for MVS, VM, and VSE users of the NetView program.

NetView/PC Publications

NetView/PC Planning, Installation, and Customization (SC31-6002) provides planning, installation, and customization information on NetView/PC and explains the communication requirements upstream to the host and downstream to supported devices. Information relating to the required PC environment and host products that support NetView/PC is also provided. It also discusses topics that are of general interest when you are ordering your equipment.

NetView/PC Application Program Interface/Communications Services Reference (SC31-6004) is a reference for OS/2 programmers who use the API/CS and for system programmers who write command processors to run under NetView. The API/CS provides a means for vendor and other external applications to use the communication services of NetView/PC.

NetView/PC Operation (SC31-6003) describes how to operate the program and diagnose problems in NetView/PC.

NetView/PC Quick Reference (SX75-0016) describes all of the functions of the F-keys throughout the NetView/PC program.

Other Network Program Products Publications

For more information about the books listed in this section, see *Bibliography and Master Index for NetView, NCP, and VTAM*.

Network Program Products General Information (GC30-3350)

Network Program Products Planning (SC30-3351)

Network Program Products Samples (SC30-3352)

Bibliography and Master Index for NetView, NCP, and VTAM (GC31-6081)²

VTAM Publications

The following list shows the books for VTAM V3R2. For information about the books for VTAM V3R1, V3R1.1, or V3R1.2, see any VTAM V3R2 book or the *Network Program Products Bibliography and Master Index*.

VTAM Installation and Resource Definition (SC23-0111)

VTAM Customization (LY30-5614)

VTAM Directory of Programming Interfaces for Customers (GC31-6403)

VTAM Operation (SC23-0113)

VTAM Messages and Codes (SC23-0114)

VTAM Programming (SC23-0115)

VTAM Programming for LU 6.2 (SC30-3400)

VTAM Diagnosis Guide (LY30-5601)

VTAM Data Areas for MVS (LY30-5592)

VTAM Data Areas for VM (LY30-5593)

VTAM Data Areas for VSE (LY30-5594)

VTAM Reference Summary (LY30-5600)

NCP, SSP, and EP Publications

The following list shows the related books for NCP V4 and NCP V5.

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP, SSP, and Related Products Directory of Programming Interfaces for Customers (GC31-6202)

NCP Migration Guide (SC30-3252 for NCP V4 and SC30-3440 for NCP V5)

² When available.

NCP, SSP, and EP Resource Definition Guide
(SC30-3349 for NCP V4 and SC30-3447 for NCP V5)

NCP, SSP, and EP Resource Definition Reference
(SC30-3254 for NCP V4 and SC30-3448 for NCP V5)

NCP and EP Reference Summary and Data Areas
(LY30-5570 for NCP V4 and LY30-5603 for NCP V5)

NCP Customization Guide (LY30-5571 for NCP V4
LY30-5606 for NCP V5)

NCP Customization Reference (LY30-5612 for NCP V4
and LY30-5607 for NCP V5)

SSP Customization (LY43-0021)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP and EP Reference (LY30-5569 for NCP V4 and
LY30-5605 for NCP V5)

Index

A

A (Alert) Statement 5
AAUCNMTD member, DSTINIT statement 24
AAUICPEX, INITMOD operand 47
AAUINLDM, INITMOD operand
 AUTHORIZ operand 48
 BUFNUM operand 48
 BUFSIZE operand 48
 BUFTYPE operand 48
 KEEPDISC operand 50
 KEEPMEM operand 51
 KEEPPIU operand 50
 KEEPRTM operand 50
 LOG operand 49
 LUCOUNT operand 51
 MAXEND operand 48
 NETID operand 50
 PERFMEM operand 50
 RTM operand 49
 RTMDISP operand 49
 SAW operand 48
 SESSTATS operand 49
 TRACELU operand 49
 TRACESC operand 49
AAUPRMLP member
 DSTINIT statement 24
 INITMOD statement 47
abnormal end value 60
ACB
 primary data set 25
 secondary data set 25
ACBNAME, VPDINIT operand 95
access method services 26
action-name operands, IF-THEN statement
 BEEP 44
 DISPLAY 44
 HCYLOG 45
 HOLD 45
 NETLOG 45
 SYSLOG 46
actions, IF-THEN statement
 execute-actions 42
 yes-or-no-actions 44
administration, system 3
AFTER command 14
alert
 focal point 22
 user 21
alias name translation 7
ALIASMEM statement 7
allocating storage 90
ALL, IF-THEN operand 43

AL, W (Wrap) operand 98
AMS
 See access method services
application name
 CNM data transfer task 19
 NetView 12, 67
APPLS, T (Timer) operand 91
ASSIGN command 9, 45
AT command 14
AUTH statement 8
AUTHDOM, AAUICPEX operand 47
authorized receiver, message 8
AUTHORIZ, AAUINLDM operand 48
AUTOFLIP, LOGINIT operand 56
automation
 message 34
 task, reinstating 60
AUTOMSG command 34
AUTOTASK command 70

B

BACKUP, DEFFOCPT operand 22
BEEP, IF-THEN operand 44
BINDFAIL, KCLASS operand 54
bitstring-function comparison-expression, IF-THEN
 operand 40
bitstring-names, IF-THEN statement
 DESC 41
 ROUTCDE 41
BNJAPAMA member 21
BNJMBDST member
 CTL statement 21
 DSTINIT statement 24
 R (Ratio) statement 81
 RATE statement 82
 REPORTS statement 83
 W (WRAP) statement 97
BNJ36DST member, DSTINIT statement 24
BOUNDS, PCLASS operand 74
buffers, VSAM 96
BUFNUM, AAUINLDM operand 48
BUFSIZE, AAUINLDM operand 48
BUFTYPE, AAUINLDM operand 48

C

C (Command List) statement 10
CDMNSESS statement 11
CDRMDEF statement 12
CDRSC, VTAM definition statement 57
character translation sets 93
class
 keep 53, 58

- class (*continued*)
 - performance 58
 - scope 13
 - service 20
- CLOSE command 27
- CMDCLASS statement 13
- CMDMDL statement 14
- CMDSYN statement 17
- CMD, IF-THEN operand 42
- CNM data transfer 18, 19
- CNMAUTH statement 18
- CNMFANGC module 30
- CNMTARG statement 19
- coding conventions viii
- command
 - defining 14
 - echoing to screen 15
 - module, loading at initialization 15
 - operands, defining synonyms 73
 - restricting usage of 13
 - run automatically at initialization 66
 - scope checking 13
 - keywords 55
 - values 94
 - synonyms, defining 17
 - type 14
- command list
 - defining 14
 - definition of viii
 - displaying 10
 - restricting usage of 13
 - run automatically at initialization 66
 - scope checking 13
 - synonyms, creating 17
- comments 4
- compare-items, IF-THEN statement
 - literals 37
 - place-holders 39
 - variable-names 38
- comparison-expressions, IF-THEN statement
 - bitstring-function 40
 - parse-function 36
 - token-function 40
- COS statement 20
- cross-domain
 - authorization 23, 47
 - sessions
 - data retrieval 12
 - global 84
 - maximum number of 11
- CTL
 - AUTH operand 8
 - CMDMDL operand 15
 - CNMAUTH operand 18
 - statement 21

D

- DASD, KCLASS operand 54
- data services command processor 25
- data services request blocks 25
- DATA, KCLASS operand 54
- DEFAULTS command 46
- DEFFOCPT statement 22
- DESC, IF-THEN operand 41
- DISPLAY, IF-THEN operand 44
- DMNPSW, NCCFID operand 67
- DOMAINID
 - IF-THEN operand 36
 - NCCFID operand 67
- DOMAINS statement 23
- domain, cross
 - See cross-domain
- DROP, NCCFID operand 67
- DSCP
 - See data services command processor
- DSIALATD member
 - ALIASMEM statement 7, 8
 - DSTINIT statement 24
- DSIALTAB member
 - COS statement 20
 - LU statement 57
 - MODE statement 64
 - ORIGNET statement 72
- DSIAMLTD member
 - CDRMDEF statement 12
 - DSTINIT statement 24
- DSICMD member
 - CMDCLASS statement 13
 - CMDMDL statement 14
 - CMDSYN statement 17
 - KEYCLASS statement 55
 - PARMSYN statement 73
 - VALCLASS statement 94
- DSICNM member
 - A (Alert) statement 5
 - C (Command List) statement 10
 - F (Filter) statement 29
 - O MONIT statement 68
 - T (Timer) statement 91
- DSICPINT member, DSTINIT statement 24
- DSICRTR, CNM router subtask 25
- DSICRTTD member
 - DEFFOCPT statement 22
 - DSTINIT statement 24
- DSICTMOD data set 27
- DSIDMN member
 - CDMNSESS statement 11
 - HARDCOPY statement 32
 - MAXABEND statement 60
 - MAXLOGON statement 61
 - MAXSPAN statement 62
 - NCCFIC statement 66
 - NCCFID statement 67

DSIDMN member (continued)

OPTIONS statement 71
POS statement 77
POSPPOOL statement 78
RRD statement 84
STORPOOL statement 90
TASK statement 92
TRANSTBL statement 93
VSAMLSR statement 96

DSIELMEM member, DSTINIT statement 24

DSIHMF member, HOLDMSG statement 33

DSILOGBK member

DSTINIT statement 24
LOGINIT statement 56

DSILUCTD member

CNMAUTH statement 18
CNMTARG statement 19
DSTINIT statement 24

DSIOPF member

OPERATOR statement 70
PROFILEN statement 80

DSIPARM data set

AAUCNMTD member, DSTINIT statement 24

AAUPRMLP member

DSTINIT statement 24
INITMOD statement 47

BNJMBDST member

CTL statement 21
DSTINIT statement 24
R (Ratio) statement 81
RATE statement 82
REPORTS statement 83
W (Wrap) statement 97

BNJ36DST member, DSTINIT statement 24

DSIALATD member

ALIASMEM statement 7
DSTINIT statement 24

DSIALTAB member

COS statement 20
LU statement 57
MODE statement 64
ORIGNET statement 72

DSIAMLTD member

CDRMDEF statement 12
DSTINIT statement 24

DSICMD member

CMDCLASS statement 13
CMDMDL statement 14
CMDSYN statement 17
KEYCLASS statement 55
PARMSYN statement 73
VALCLASS statement 94

DSICNM member

A (Alert) statement 5
C (Command List) statement 10
F (Filter) statement 29
O MONIT statement 68
T (Timer) statement 91

DSIPARM data set (continued)

DSICPINT member, DSTINIT statement 24

DSICRRTD member

DEFFOCPT statement 22
DSTINIT statement 24

DSIDMN member

CDMNSESS statement 11
HARDCOPY statement 32
MAXABEND statement 60
MAXLOGON statement 61
MAXSPAN statement 62
NCCFIC statement 66
NCCFID statement 67
OPTIONS statement 71
POS statement 77
POSPPOOL statement 78
RRD statement 84
STORPOOL statement 90
TASK statement 92
TRANSTBL statement 93
VSAMLSR statement 96

DSIELMEM member, DSTINIT statement 24

DSIHMF member, HOLDMSG statement 33

DSILOGBK member

DSTINIT statement 24
LOGINIT statement 56

DSILUCTD member

CNMAUTH statement 18
CNMTARG statement 19
DSTINIT statement 24

DSIOPF member

OPERATOR statement 70
PROFILEN statement 80

DSISPN member, SPANLIST statement 86

DSITRCBK member

DSTINIT statement 24
LOGINIT statement 56

DSIVPARM member, VPDINIT statement 95

IF-THEN automation statement 34

KEEPMEM member

KCLASS statement 53
MAPSESS statement 58

PERFMEM member

MAPSESS statement 58
PCLASS statement 74

USRSQMEM member, DSTINIT statement 24

VSAM members

AAUCNMTD 24
AAUPRMLP 24
BNJMBDST 24
BNJ36DST 24
DSIALATD 24
DSIAMLTD 24
DSICPINT 24
DSICRRTD 24
DSIELMEM 24
DSILOGBK 24
DSILUCTD 24
DSITRCBK 24

DSIPARM data set (*continued*)
 VSAM members (*continued*)
 USRSQMEM 24

DSIPRF data set
 DSIPROFA member
 AUTH statement 8
 DOMAINS statement 23
 ISPAN statement 52
 PROFILE statement 79
 SPAN statement 85
 DSIPROFB member
 OPCLASS statement 69
 PROFILE statement 79

DSIPROFA member
 AUTH statement 8
 DOMAINS statement 23
 ISPAN statement 52
 PROFILE statement 79
 SPAN statement 85

DSIPROFB member
 OPCLASS statement 69
 PROFILE statement 79

DSISPN member, SPANLIST statement 86

DSITRCBK member
 DSTINIT statement 24
 LOGINIT statement 56

DSIVPARM member, VPDINIT statement 95

DSIVTAM data set
 minor node definition statements 63
 STATOPT statement 87

DSIWLS macro 26

DSPLYLOC, PCLASS operand 74

DSRB
 See data services request blocks

DSRBO, DSTINIT operand 25

DSRBU, DSTINIT operand 25

DST initialization
 exit routine 12
 AAUINIT, defined as 47
 DSIAINIT, defined as 7
 DSILINIT, defined as 12, 18, 19
 parameters 24
 DSTINIT statement 24

E

EBCDIC character set 93

ECHO, DSTINIT operand 15

END statement 4

error-to-traffic ratio value 81

ER, MAPSESS operand 59

event logging 82

EVERY command 14

EV, W (Wrap) operand 97

execute-actions 42

EXEC, IF-THEN operand 42

exit routine 12, 47

external log record formats
 hardware monitor 101
 session monitor 108
 accounting and availability data collection record 109
 bind failure record 109
 combined session start-end record 109
 init failure record 110
 RTM collection record 108
 session end record 108
 session start record 109
 storage and event counter record 110
 subvector formats 111

F

F (Filter) statement 29

FAILURES, KCLASS operand 54

focal point nodes, defining 22

FORCE command 54

FUNCT, DSTINIT operand 24

H

hard-copy printers, defining 32

HARDCOPY statement 32

hardware monitor
 data base
 event logging 82
 external log formats 101
 E/T ratios, specifying 97
 wrap counts, specifying 97
 focal point nodes 22
 report record logging 83

HCL, PROFILE operand 79

HCYLOG, IF-THEN operand 45

HDRMTYPE, IF-THEN operand 36

help panels 17

HOLDMSG statement 33

HOLD, IF-THEN operand 45

I

IC
 NCCFIC operand 66
 PROFILE operand 79

IF-THEN automation statement 34

INITFAIL, KCLASS operand 54

initialization
 command list, running automatically at 66
 command module, loading at 15
 command, running automatically at 66
 parameters
 DST 24
 Netview sequential log 56
 network log 56
 trace log 56
 VPDTASK 95

initialization (*continued*)
 record logging at 83
INITMOD statement 47
INIT, TASK operand 92
installation parameters 47
interconnected network 20, 57
ISPAN statement 52

J

Japanese, character translation 93
JOBNAME, IF-THEN operand 36

K

Kanji, character set 93
Katakana, character set 93
KCLASS
 MAPSESS operand 58
 statement 53
keep class
 defining 53
 mapping sessions into 58
KEEP PIUS command 50
KEEPDISC, AAUINLDM operand 50
KEEPMEM member 51
 KCLASS statement 53
 MAPSESS statement 58
KEEPIU
 AAUINLDM operand 50
 KCLASS operand 54
KEEPRTM, AAUINLDM operand 50
KEYCLASS statement 55

L

labels 4
LBUILD macro 87
literal compare-item, IF-THEN statement 37
local shared resources 27
logical unit (LU) name
 See LU
LOGINIT statement 56
logon
 attempts, specifying maximum 61
 mode names 64
 security 71
LOG, AAUINLDM operand 49
LU
 CNMTARG operand 19
 names
 defining 19
 translating 57
 number in network 51
 session authorization 18, 19
 statement 57
LUCOUNT, AAUINLDM operand 51

M

MACRF, DSTINIT operand 27
MAPSESS statement 58
MAXABEND statement 60
MAXEND, AAUINLDM operand 48
MAXLOGON statement 61
MAXSESS, CNMAUTH operand 18
MAXSPAN statement 62
member, definition of viii
MEM, TASK operand 92
message
 authorized receiver 8
 automation 34
 buffer, parsing 16
 holding on operator screen 33
 indicator classes 5, 29
 defining alarm attribute 5
 defining color attribute 5
 defining highlighting attribute 5
 unsolicited 8
MOD
 CMDMDL operand 14
 TASK operand 92
 TRANSTBL operand 93
mode names, logon 64
MODE statement 64
modem configuration change form 137
monitor
 hardware
 See hardware monitor
 session
 See session, monitor
 status
 See status monitor
MOVE command 60
MSGCMD statement 65
MSGID, IF-THEN operand 37
MSGRECV, AUTH operand 8
MSGROUTE command 42
MVS, definition of viii

N

NCCFIC statement 66
NCCFID statement 67
NETID, AAUINLDM operand 50
NETLOG, IF-THEN operand 45
NetView
 application name 12, 67
 sequential log 56
network
 interconnected 20
 log 5
 initial parameters 56
 placing messages in 45
 name, defining 72

network asset management
 sample command list record formats
 DCE hardware subrecords 135
 end subrecords 133
 error subrecords 136
 PU hardware subrecords 134
 PU software subrecords 134
 start subrecords 132
 subrecord prefix 132
 timeout subrecords 136
 vital product data descriptions
 answering node configuration data 123
 attached device configuration data 130
 DCE data for DSUs/CSUs 127
 DCE data for modems 126
 link configuration data 128
 LPDA2 sense data 129
 product data 124
 node
 focal point 22
 major, defining 89
 minor, defining 63
 reactivation 68
 VTAM 86, 87

O

O MONIT statement 68
 OBJPCT, PCLASS operand 76
 OBJTIME, PCLASS operand 74
 OFF, REPORTS operand 83
 ONE, IF-THEN operand 43
 ON, REPORTS operand 83
 OPCLASS statement 69
 operands
 AAUICPEX, INITMOD statement 47
 AAUINLDM, INITMOD statement
 AUTHORIZ 48
 BUFNUM 48
 BUFSIZE 48
 BUFTYPE 48
 KEEPDISC 50
 KEEPMEM 51
 KEEPPIU 50
 KEEPRTM 50
 LOG 49
 LUCOUNT 51
 MAXEND 48
 NETID 50
 PERFMEM 50
 RTM 49
 RTMDISP 49
 SAW 48
 SESSTATS 49
 TRACELU 49
 TRACESC 49
 ACBNAME, VPDINIT statement 95
 ALL, IF-THEN statement 43

operands (continued)
 AL, W (Wrap) statement 98
 APPLS, T (Timer) Statement 91
 AUTHDOM, INITMOD statement 47
 AUTOFLIP, LOGINIT statement 56
 BACKUP, DEFFOCPT statement 22
 BEEP, IF-THEN statement 44
 BINDFAIL, KCLASS statement 54
 BOUNDS, PCLASS statement 75
 CMD, IF-THEN statement 42
 CTL
 AUTH statement 8
 CMDMDL statement 15
 CNMAUTH statement 18
 DASD, KCLASS statement 54
 DATA, KCLASS statement 54
 defining synonyms for 73
 DESC, IF-THEN statement 41
 DISPLAY, IF-THEN statement 44
 DMNPSW, NCCFID statement 67
 DOMAINID
 IF-THEN statement 36
 NCCFID statement 67
 DROP, NCCFID statement 67
 DSPLYLOC, PCLASS statement 75
 DSRBO, DSTINIT statement 25
 DSRBU, DSTINIT statement 25
 ECHO, CMDMDL statement 15
 ER, MAPSESS statement 59
 EV, W (Wrap) statement 97
 EXEC, IF-THEN statement 42
 FAILURES, KCLASS statement 54
 FUNCT, DSTINIT statement 24
 HCL, PROFILE statement 79
 HCYLOG, IF-THEN statement 45
 HDRMTYPE, IF-THEN statement 36
 HOLD, IF-THEN statement 45
 IC
 NCCFIC statement 66
 PROFILE statement 79
 INITFAIL, KCLASS statement 54
 INIT, TASK statement 92
 JOBNAME, IF-THEN statement 36
 KCLASS, KCLASS statement 58
 KEEPPIU, KCLASS statement 54
 LU, CNMTARG statement 19
 MACREF, DSTINIT statement 27
 MAXSESS, CNMAUTH statement 18
 MEM, TASK statement 92
 MOD
 CMDMDL statement 14
 TASK statement 92
 TRANSTBL statement 93
 MSGID, IF-THEN statement 37
 MSGRECV, AUTH statement 8
 NETLOG, IF-THEN statement 45
 OBJPCT, PCLASS statement 76
 OBJTIME, PCLASS statement 75

operands (*continued*)

OFF, REPORTS statement 83
ONE, IF-THEN statement 43
ON, REPORTS statement 83
OTHER
 KEYCLASS statement 55
 VALCLASS statement 94
PARSE, CMDMDL statement 16
PASSWORD
 OPERATOR statement 70
 VPDINIT statement 95
PBSDN, DSTINIT statement 24
PCLASS, MAPSESS statement 58
PDDNM, DSTINIT statement 25
PENDING, T (Timer) Statement 91
PERSIST
 CNMTARG statement 19
 DSTINIT statement 27
PPASS, DSTINIT statement 25
PPT, IF-THEN statement 43
PRI
 MAPSESS statement 58
 TASK statement 92
PRIMARY, DEFFOCPT statement 22
RESUME, LOGINIT statement 56
RES, CMDMDL statement 15
RETRY
 CDRMDEF statement 12
 DSTINIT statement 27
ROUTCDE, IF-THEN statement 41
ROUTE, IF-THEN statement 42
RTDEF, PCLASS statement 75
RTMDATA, KCLASS statement 54
SAW, KCLASS statement 53
SBSDN, DSTINIT statement 25
SDDNM, DSTINIT statement 25
SEC, MAPSESS statement 59
SESSFAIL, KCLASS statement 54
SESSID, IF-THEN statement 37
SESSNORM, KCLASS statement 54
SIZE, STORPOOL statement 90
SNAPRQ, VPDINIT statement 95
SPAN, minor node definitions 63
SPASS, DSTINIT statement 25
ST, W (Wrap) statement 97
SUPPCHAR, NCCFID statement 67
SYSID, IF-THEN statement 37
SYSLOG, IF-THEN statement 46
TEXT, IF-THEN statement 37
TOKEN, IF-THEN statement 40
TP, MAPSESS statement 59
TRACDATA, KCLASS statement 54
TSKID, TASK statement 92
TYPE, CDRMDEF statement 14
UNSOL, DSTINIT statement 25
VERIFY, OPTIONS statement 71
VPDREQ, VPDINIT statement 95
VPDSTOR, VPDINIT statement 95

operands (*continued*)

VPDWAIT, VPDINIT statement 95
VR, MAPSESS statement 59
XITBN, DSTINIT statement 26
XITBO, DSTINIT statement 26
XITCI, DSTINIT statement 26
XITCO, DSTINIT statement 26
XITDI, DSTINIT statement 26
XITVI, DSTINIT statement 26
XITVN, DSTINIT statement 26
XITVO, DSTINIT statement 26
XITXL, DSTINIT statement 26
OPERATOR
 authority, defining 8
 autotask 9
 cross-domain sessions 23, 84
 logon 70
 attempts, specifying maximum 61
 security 71
 profile 23, 79, 80
 routing messages to 8
 scope of commands 69
 span of control
 NetView 8
 VTAM 85
 statement 70
OPTIONS statement 71
ORIGNET statement 72
OTHER
 KEYCLASS operand 55
 VALCLASS operand 94
OVERRIDE command 46

P

PARMSYN statement 73
parse-function comparison-expressions 36
parse-names
 DOMAINID 36
 HDRMTYPE 36
 JOBNAME 36
 MSGID 37
 SESSID 37
 SYSID 37
 TEXT 37
PARSE, CMDMDL operand 16
PASSWORD
 OPERATOR operand 70
 VPDINIT operand 95
PBSDN, DSTINIT operand 24
PCCU macro 87
PCLASS
 MAPSESS operand 58
 statement 74
PDDNM, DSTINIT operand 25
PENDING, T (Timer) operand 91
PERFMEM
 AAUINLDM operand 50

PERFMEM (continued)

- MAPSESS statement 58
- PCLASS statement 74
- performance class, mapping sessions into 58
- PERSIST
 - CNMTARG operand 19
 - DSTINIT operand 27
- persistent sessions 19, 27
- place-holder compare-item 39
- POS statement 77
- POSPPOOL statement 78
- PPASS, DSTINIT operand 25
- PPT
 - IF-THEN operand 43
 - routing messages to 43
- PRI
 - MAPSESS operand 58
 - TASK operand 92
- PRIMARY, DEFFOCPT operand 22
- PROFILE
 - defining 79
 - operator 80
 - statement 79
- PROFILEN statement 80

R

- R (Ratio) statement 81
- RATE statement 82
- ratio, error-to-traffic (E/T) 81
- report record logging 83
- REPORTS statement 83
- RESET IMMED command 60
- response time monitor 49, 74
- RESUME, LOGINIT operand 56
- RES, CMDMDL operand 15
- RETRY
 - CDRMDEF operand 12
 - DSTINIT operand 27
- REXX, definition of viii
- ROUTCDE, IF-THEN operand 41
- ROUTE, IF-THEN operand 42
- RRD statement 84
- RTDEF, PCLASS operand 74
- RTMDATA, KCLASS operand 54
- RTMDISP, AAUINLDM operand 49
- RTM, AAUINLDM operand 49

S

- sample command list record formats, network asset management
 - See network asset management
- SAW
 - AAUINLDM operand 48
 - KCLASS operand 53
- SBSDN, DSTINIT operand 25

- scope checking
 - command lists 13, 14
 - commands 13, 14, 69
 - keywords 55, 139
 - values 94, 139
- SDDNM, DSTINIT operand 25
- security, operator 71
- SEC, MAPSESS operand 59
- service, class of 20
- SESSFAIL, KCLASS operand 54
- SESSID, IF-THEN operand 37
- session
 - awareness 48
 - CNM data transfer 18
 - cross-domain 11, 12
 - ending 67
 - LU, authorization 18, 19
 - monitor 12
 - conversations 12, 27
 - external log record formats 108
 - installation parameters 47
 - session data retrieval 12
 - operator, reinstating 60
 - parallel 18
 - persistent 19
 - TAF 29
 - terminal, reinstating 60
- SESSNORM, KCLASS operand 54
- SESSTATS, AAUINLDM operand 49
- SIZE, STORPOOL operand 90
- SNAPRQ, VPDINIT operand 95
- SPAN
 - minor node definition operand 63
 - names, number allowed 62
 - statement 85
 - VTAM 85, 86
- SPANLIST statement 86
- SPASS, DSTINIT operand 25
- SSCP 12
 - See also system services control point (SSCP)
- SSI 37
 - See also subsystem interface
- START command 60
- STATOPT statement 87
- status monitor
 - displaying commands or command lists 10
 - functions 87
 - important message indicator 45
 - reactivating nodes 68
- STOP command 52, 60
- storage allocation 90
- STORPOOL statement 90
- ST, W (Wrap) operand 97
- subsystem interface 37
- SUPPCHAR, NCCFID operand 67
- suppressing output 67
- synonyms, defining
 - command lists 17

synonyms, defining (*continued*)
 commands 17
 operands 73
syntax conventions 4
SYSID, IF-THEN operand 37
SYSLOG, IF-THEN operand 46
system services control point (SSCP)
 cross-domain, name 12
 information, tracing 49
 LUs owned by 51
 session awareness data for 53

T

T (Timer) statement 91
TAF sessions 29
TASK statement 92
tasks, defining 92
terminals, defining 77, 78
TEXT, IF-THEN operand 37
timers, controlling VTAM commands with 91
token-function comparison-expressions 40
TOKEN, IF-THEN operand 40
TP, MAPSESS operand 59
TRACDATA, KCLASS operand 54
trace
 data 48, 50
 log 56
TRACELU, AAUINLDM operand 49
TRACESC, AAUINLDM operand 49
translation
 alias name 7
 character 93
TRANSTBL statement 93
TSKID, TASK operand 92
TYPE, CMDMDL statement 14

U

unsolicited messages, routing to an operator 8
UNSOL, DSTINIT operand 25
USRSQMEM member, DSTINIT statement 24

V

VALCLASS statement 94
variable-name compare-item 38
VBUILD macro 87
VERIFY, OPTIONS operand 71
vital product data descriptions, network asset management
 See network asset management
VM, definition of viii
VPDINIT statement 95
VPDREQ, VPDINIT operand 95
VPDSTOR, VPDINIT operand 95
VPDTASK NetView subtask 95

VPDWAIT, VPDINIT operand 95
VR, MAPSESS operand 59
VSAM
 buffers 96
 file, mapping sessions to 54
 members
 AAUCNMTD member 24
 AAUPRMLP member 24
 BNJMBDST member 24
 BNJ36DST member 24
 DSIALATD member 24
 DSICPINT member 24
 DSIELMEM member 24
 DSILOGBK member 24
 DSILUCTD member 24
 DSITRCBK member 24
 USRSQMEM member 24
 services 25
VSAMLSR statement 96
VTAM
 CDRCS definition statement 57
 commands, controlling with timers 91
 definition of viii
 SPAN operand 63
 spans 85, 86

W

W (Wrap) statement 97
wrap count
 initial 21, 97
 statements 21

X

XITBN, DSTINIT operand 26
XITBO, DSTINIT operand 26
XITCI, DSTINIT operand 26
XITCO, DSTINIT operand 26
XITDI, DSTINIT operand 26
XITVI, DSTINIT operand 26
XITVN, DSTINIT operand 26
XITVO, DSTINIT operand 26
XITXL, DSTINIT operand 26

Y

yes-or-no-actions 44

Reader's Comment Form

NetView™
Administration Reference
Release 3

Publication No. SC31-6014-0

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM Publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval, and legibility.

Comments: _____

What is your occupation? _____

If you wish a reply, give your name, company, mailing address, and date:

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E15
P.O. Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and tape

Please Do Not Staple

Fold and tape



